

Trabajo Fin de Grado

Grado en Ingeniería de las Tecnologías de Telecomunicación

Detección automática de colores en imágenes dermatoscópicas de lesiones pigmentadas

Autora: Julia Moreno Casanova

Tutoras: Begoña Acha Piñero y Carmen Serrano Gotarredona

Dpto. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2018



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Detección automática de colores en imágenes dermatoscópicas de lesiones pigmentadas

Autora:

Julia Moreno Casanova

Tutoras:

Begoña Acha Piñero

(Catedrática de Universidad)

Carmen Serrano Gotarredona

(Catedrática de Universidad)

Dpto. de Teoría de la Señal y Comunicaciones

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2018

Trabajo Fin de Grado: Detección automática de colores en imágenes dermatoscópicas de lesiones pigmentadas

Autora: Julia Moreno Casanova

Tutoras: Begoña Acha Piñero y Carmen Serrano Gotarredona

El tribunal nombrado para juzgar el Trabajo arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal

A mi familia

Agradecimientos

A todas las personas que me han ayudado a lo largo de los años.

Julia Moreno Casanova

Sevilla, 2018

Resumen

En este Trabajo Fin de Grado se expone una técnica que se encarga de la detección automática de colores en imágenes dermatoscópicas de lesiones pigmentadas, basada en un algoritmo orientado a la identificación de pigmentos en este tipo de imágenes, mediante una nueva metodología fundamentada en el empleo de modelos de mezcla Gaussiana. Dicha detección de los colores presentes en las lesiones pigmentadas de imágenes dermatoscópicas es muy importante para determinar el diagnóstico de malignidad de cada una de ellas. Este algoritmo se propone en el artículo: Color identification in dermoscopy images using Gaussian mixture models, escrito por Barata, C., Figueiredo, M. A. T., Celebi, M. E., & Marques, J. S. [1].

El primer paso que lleva a cabo la técnica propuesta en este documento, es el del cálculo de 6 modelos de mezcla Gaussianas, uno para cada uno de los colores dermatoscópicos. Para este propósito, se hace uso de un conjunto de parches, que servirán para el debido entrenamiento del algoritmo. Una vez calculadas las 6 mezclas, ya se puede pasar a la detección y cuantificación de colores para las imágenes que forman parte de la base de datos.

Finalmente, los resultados muestran que el método propuesto no es muy eficiente en cuanto a la asignación de colores a los píxeles que componen las lesiones pigmentadas analizadas. Esto se comprueba mediante los parámetros de Sensibilidad (Sen (detección de colores) = 69,40 % y Sen (asignación de colores a los píxeles de las lesiones) = 44,06%), VPP (VPP (detección de colores) = 77,91% y VPP (asignación de colores a los píxeles de las lesiones) = 29,84%) y Exactitud (Acc (detección de colores) = 75,66% y Acc (asignación de colores a los píxeles de las lesiones) = 90,03%).

Abstract

This Bachelor Thesis, shows a technique which undertakes the automatic detection of colors in dermoscopic images of pigmented skin lesions. This technique is based in an algorithm oriented to the identification of colors in this type of images, using a new methodology based on the utilization of Gaussian mixture models. That detection of the colors present in the pigmented lesions of the dermoscopic images, is very important in order to determine the malignancy diagnosis of each lesion. This algorithm is proposed in the article: Color identification in dermoscopy images using Gaussian mixture models, written by Barata, C., Figueiredo, M. A. T., Celebi, M. E., & Marques, J. S. [1].

The first step carried out by the proposed technique, is the calculation of 6 Gaussian mixture models, one for each dermoscopic color. For that purpose, a set of patches is used as a training set for the algorithm. Once the six mixtures have been calculated, it is time to carry out the detection and quantification of colors for the images that belongs to the data base.

Finally, the results show that the proposed method is not very efficient as to the pixel color label evaluation performed for every analyzed lesion. This is proved by parameters like Sensitivity (Sen (colors detection) = 69,40 % y Sen (pixel color label evaluation) = 44,06%), PVV (VPP (colors detection) = 77,91% y VPP (pixel color label evaluation) = 29,84%) y Accuracy (Acc (colors detection) = 75,66% y Acc (pixel color label evaluation) = 90,03%).

Índice

Agradecimientos	i
Resumen	iii
Abstract	v
Índice	vii
Índice de Tablas	ix
Índice de Figuras	xi
Notación	xv
1 Introducción	1
2 Estado del Arte	3
3 Metodología	5
3.1. <i>Modelo de aprendizaje de la mezcla de color</i>	6
3.1.1. Modelos de finitos de mezclas	7
3.1.2. Criterio propuesto	8
3.1.3. Algoritmo utilizado para la implementación del criterio propuesto	11
3.2. <i>Asignación y cuantificación del color</i>	13
3.2.1. Procedimiento para la asignación del color	14
3.2.2. Procedimiento para la cuantificación del color	14
4 Resultados	15
4.1. <i>Base de datos de imagen</i>	15
4.2. <i>Ajuste y configuración de parámetros del código</i>	16
4.2.1. Parámetros de la función Color_Mixture_Palette	16
4.2.2. Parámetros de las funciones Detect_Colors y Quantify_Colors	16
4.3. <i>Resultados de la identificación de colores en las lesiones</i>	17
4.3.1. Obtención de las máscaras de segmentación de las imágenes evaluadas	17
4.3.2. Evaluación de la asignación de colores a los píxeles de las lesiones	22
4.3.3. Evaluación de la detección de colores en las lesiones	34
4.3.4. Comparación del algoritmo propuesto con otro método	37
5 Conclusiones	41
5.1. <i>Conclusiones extraídas de los resultados experimentales</i>	41
6 Mejoras y Líneas Futuras	43
6.1. <i>Mejoras del algoritmo propuesto</i>	43
6.1.1. Empleo de la constancia del color para un mejor análisis de imágenes dermatoscópicas	43

6.2. Líneas Futuras: Aplicación de la detección de color para una posterior clasificación de la malignidad y benignidad de las lesiones.	45
6.1.1. Empleo de un sistema inteligente de soporte a decisiones para la detección del cáncer de piel a partir de imágenes dermatoscópicas	45
6.2.2. Empleo de una técnica de reconstrucción 3D para la clasificación automática de lesiones dermatoscópicas en la piel	46
6.2.3. Empleo de una técnica de extracción de características de color local basada en superpixel para la clasificación de lesiones cutáneas	47
6.2.4. Empleo de la constancia del color para un mejor análisis de imágenes dermatoscópicas	47

Referencias	49
-------------	----

Anexo A: Códigos	53
A.1. Código 1	53
A.2. Código 2	55
A.3. Código 3	59
A.4. Código 4	60
A.5. Código 5	69
A.6. Código 6	70
A.7. Código 7	70
A.8. Código 8	71
A.9. Código 9: Funciones creadas para el entrenamiento del algoritmo	72
A.10. Código 10: Fichero 1	75
A.11. Código 11	75
A.12. Código 12: Fichero 2	76
A.13. Código 13: Fichero 3	76
A.14. Código 14: Fichero 4	77

Anexo B: Datos Relevantes	79
B.1. Tabla 1: Colores detectatos por el algoritmo	79
B.2. Tabla 2: Valores totales de los parámetros TP, TN, FP y FN , y para cada uno de los colores. Valores para obtener Tabla 4-1.	83
B.2.1. Color Azul-Gris	83
B.2.2. Color Marrón Oscuro	86
B.2.3. Color Marrón Claro	90
B.2.4. Color Negro	93
B.2.5. Color Rojo	97
B.2.6. Resultados totales	100
B.3. Tabla 3: Valores totales del parámetro TP, de cada color, para el caso de clasificación errónea dado en la Tabla 4-2.	100
B.4. Tabla 4: Colores detectatos por el dermatólogo	104
B.5. Tabla 5: Valores totales de las variables TP, TN, FP y FN de cada color, para el caso de la detección de colores en la Tabla 4-3.	108

ÍNDICE DE TABLAS

Tabla 4-1. VPP, Sen, Acc de la asignación de colores a los píxeles de las lesiones pigmentadas, en tanto por uno	24
Tabla 4-2. Clasificación errónea de los colores de los píxeles de imágenes dermatoscópicas. Los valores presentes en esta tabla están en términos de porcentaje.	34
Tabla 4-3. VPP, Sen, Acc de la detección de los colores dermatoscópicos de las 127 imágenes en tanto por uno.	35
Tabla 6-1. Resultados de la clasificación para la base de datos EDRA (múltiples fuentes), con la corrección de la constancia del color y sin ella.	45
Tabla 6-2. Resultados de la clasificación para la base de datos PH ² (fuente única), con la corrección de la constancia del color y sin ella.	45

ÍNDICE DE FIGURAS

Figura 1-1. Ejemplos de imágenes dermatoscópicas.	2
Figura 3-1. Esquema de la primera parte del método propuesto: Aprendizaje de modelos de mezcla Gaussiana	5
Figura 3-2. Esquema de la segunda parte del método propuesto: Asignación (a) y cuantificación (b) de colores	6
Figura 4-1. Ejemplo de los parches de los 6 colores dermatoscópicos: (a) azul-gris, (b) marrón oscuro, (c) marrón claro, (d) negro, (e) blanco, (f) rojo.	15
Figura 4-2. Interfaz ampliada de la aplicación Color Thresholder en la que se numeran todas las secciones de las que está compuesta.	18
Figura 4-3. Importación de la imagen Ael510_pixel.bmp2 de la clase Blue nevus que contiene los colores detectados por el dermatólogo que forman las máscaras de segmentación.	18
Figura 4-4. Selección del espacio de color RGB.	19
Figura 4-5. Entorno de trabajo final, listo para la segmentación de la imagen en los colores detectados.	19
Figura 4-6. Proceso de selección del color azul a través de la herramienta Select Colors, para su posterior segmentación.	20
Figura 4-7. Proceso de separación del color azul del resto de colores a través de la herramienta Find Thresholds.	20
Figura 4-8. Proceso de obtención de la máscara de segmentación binaria a través de la herramienta Show Binary.	21
Figura 4-9. Exportación de la máscara binaria del color azul al entorno de trabajo de Matlab.	21
Figura 4-10. Proceso de guardado de imágenes a partir del entorno de trabajo de Matlab.	22
Figura 4-11. Proceso de segmentación completa de la imagen Ael510 de la clase Blue nevus: (a) Imagen original, (b) Imagen con las máscaras de color detectadas por el dermatólogo, (c) Imagen de la segmentación del color azul presente en la imagen (b), (d) Imagen de la segmentación del color marrón claro presente en la imagen (b), (e) Imagen de la segmentación del color marrón oscuro presente en la imagen (b).	22
Figura 4-12. Esquema de variables TP, TN, FP, FN	23
Figura 4-13. Ejemplo de la segmentación de la lesión Ael446 de la clase Blue nevus perteneciente a la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color azul de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color azul realizada por el algoritmo. Se observa como (d) abarca mayor área que (c).	25
Figura 4-14. Comparación entre la evaluación del algoritmo y la evaluación del especialista en lo referente a la presencia del color negro en las lesiones analizadas (Gdl037 perteneciente a la clase Melanoma, Ggl006 y Nml089 pertenecientes a la clase Blue nevus y Aal012 perteneciente a la clase Clark nevus). Todas las lesiones presentadas en la imagen muestran la alta presencia del negro por encima de la detección de otro color (Gdl037, Aal012), no coincidiendo así con la la detección realizada por el especialista.	25
Figura 4-15. Comparación y detección del color negro en la lesión Nml089 de la clase Blue nevus de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color azul de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color azul realizada por el algoritmo, (e) Máscara de segmentación del color negro realizada por el algoritmo y no detectado por el especialista.	25

Figura 4-16. Comparación y detección del color negro en la lesión Aal010 de la clase Clark nevus de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color azul de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color marrón oscuro de (b) realizada por el dermatólogo, (e) Máscara de segmentación del color negro de (b) realizada por el dermatólogo, (f) Máscara de segmentación del color azul realizada por el algoritmo, (g) Máscara de segmentación del color marrón oscuro realizada por el algoritmo, (h) Máscara de segmentación del color negro realizada por el algoritmo, presentando una mayor área que (e). 26

Figura 4-17. Comparación y detección del color negro en la lesión Ngl021 de la clase Melanoma de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color azul de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color marrón oscuro de (b) realizada por el dermatólogo, (e) Máscara de segmentación del color negro de (b) realizada por el dermatólogo, (f) Máscara de segmentación del color azul realizada por el algoritmo, (g) Máscara de segmentación del color marrón oscuro realizada por el algoritmo, (h) Máscara de segmentación del color negro realizada por el algoritmo, presentando una mayor área que (e). 26

Figura 4-18. Parches antiguos del color azul-gris. 27

Figura 4-19. Sobresegmentación del color azul-gris en la lesión Ael490 perteneciente a clase Melanoma de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color azul-gris de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color azul-gris realizada por el algoritmo. 28

Figura 4-20. Subsegmentación del color azul-gris en la lesión Fgl063 perteneciente a clase Blue nevus de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color azul-gris de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color azul-gris realizada por el algoritmo. 28

Figura 4-21. Sobresegmentación del color marrón oscuro en la lesión Fhl066 perteneciente a clase Clark nevus de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color marrón oscuro de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color marrón oscuro realizada por el algoritmo. 29

Figura 4-22. Subsegmentación del color marrón oscuro en la lesión Acl039 perteneciente a clase Clark nevus de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color marrón oscuro de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color marrón oscuro realizada por el algoritmo. 29

Figura 4-23. Sobresegmentación del color marrón claro en la lesión nfl072 perteneciente a clase Melanoma de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color marrón claro de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color marrón claro realizada por el algoritmo. 30

Figura 4-24. Subsegmentación del color marrón claro en la lesión Acl283 perteneciente a clase Clark nevus de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color marrón claro de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color marrón claro realizada por el algoritmo. 30

Figura 4-25. Imágenes (Nbl042,Ngl041,nal095,de arriba hacia abajo) analizadas para el color rojo pertenecientes a la clase Melanoma de la base de datos. Conjunto de imágenes: (a),(e),(i) Imágenes originales,(b),(f),(j) Versiones clasificadas de (a),(e),(i) ,respectivamente, por el dermatólogo, (c),(g),(k) Máscaras de segmentación del color rojo de (b),(f),(j) realizada por el dermatólogo, (d) ,(h),(l) Máscaras de segmentación del color rojo realizada por el algoritmo. 31

Figura 4-26. Evaluación de la imagen FBL032 de la clase Blue nevus de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Clasificación realizada por el algoritmo, (c) Clasificación realizada por el dermatólogo. 32

Figura 4-27. Ejemplos de la detección de colores en lesiones completas. Conjunto de imágenes: (a),(b),(c),(d) Imágenes originales, (e),(f),(g),(h) Versiones clasificadas realizadas por el algoritmo. 38

Figura 4-28. Ejemplos de la evaluación del dermatólogo: (a) Imagen original, (b) Detección de color de (a) realizada por el algoritmo, (c) Imagen evaluada por el dermatólogo en la que las zonas verdes simbolizan las regiones mal clasificadas (esa región debería haber sido marrón clara en vez de azul). (d) Imagen original, (e)

Detección de color de (d) realizada por el algoritmo, (f) Imagen evaluada por el dermatólogo en la que las zonas verdes simbolizan las regiones mal clasificadas (de acuerdo con el dermatólogo no hay marrón claro). (g) Imagen original, (h) Detección de color de (g) realizada por el algoritmo, (i) Imagen evaluada por el dermatólogo en la que las zonas verdes simbolizan las regiones mal clasificadas (esa región mal clasificada corresponde a piel saludable que se ha incluido en la lesión debido a la segmentación realizada por el algoritmo) [4] 39

Figura 6-1. Ejemplos de imágenes obtenidas bajo diferentes condiciones de iluminación. Conjunto de imágenes: (a) y (b) son melanomas, (c) y (d) son lesiones benignas 43

Figura 6-2. Ejemplos de constancia del color. Conjunto de imágenes: (a),(b),(c),(d),(e),(f) Imágenes originales, (g),(h),(i),(j),(k),(l), Imágenes obtenidas mediante Shade of Gray $p=6$. Las tres primeras columnas son melanomas, mientras que las tres últimas son lesiones benignas. 44

Figura 6-3. Imagen de seguimiento y proyecciones 3-D de la profundidad de un melanoma. Conjunto de imágenes: (a) Imagen de seguimiento (imagen superior) y profundidad estimada (imagen inferior), (b) Región de interés del melanoma (imagen superior) y profundidad estimada de la misma (imagen inferior) [33]. 46

$\arg \max_{\theta}\{A\}$	Argumento máximo de A
$\arg \min_{\theta}\{A\}$	Argumento mínimo de A
block	Bloque diagonal
diag	Diagonal
et al.	Y otros
exp	Función exponencial
$E(A)$	Esperanza del suceso A
log	Logaritmo en base 10
$\Pr(A)$	Probabilidad del suceso A
$<$	Menor
$>$	Mayor
\geq	Mayor o igual
\propto	Proporcionalidad
\in	Pertenece a

1 INTRODUCCIÓN

La felicidad radica, ante todo, en la salud

- George W. Curtis-

En los últimos años, el reconocimiento de las lesiones producidas en la piel se ha convertido en un tema de actualidad que ha atraído la atención de varios sectores importantes de la población. Es sabido, como se relata en [2], que en el año 2012 una gran cantidad de tipos de cáncer de piel, entre los que destaca el melanoma maligno por su agresividad, afectaron a unas 231000 personas causando la muerte a 55000 de ellas. La mayoría de estas muertes tuvieron lugar en Nueva Zelanda y Australia, donde gran parte de la población es de etnia Caucásica, y como bien ilustran los datos históricos, desde hace 20 años el melanoma maligno es más probable que se dé en zonas en las que habita este tipo de etnia. Al igual que todos los cánceres existentes, el melanoma maligno puede extenderse por otros órganos del cuerpo, lo que se denomina como metástasis. Esto ocurre cuando la célula de la piel, formada por melanocitos, muta a una clase de célula cancerígena que contiene pigmento en la piel [2]. Por tanto, un diagnóstico a tiempo del melanoma es muy importante para conseguir un tratamiento efectivo de la enfermedad.

Una de las técnicas más usadas por los dermatólogos y médicos de hoy en día, para un mejor diagnóstico de lesiones en la piel, y en consecuencia de melanomas, es la llamada dermatoscopia. La dermatoscopia es un método no invasivo empleado para observaciones *in vivo* de lesiones producidas en la piel. Según [3], para poder llevar a cabo este método, el especialista emplea un instrumento de aumento (dermatoscopio, estereomicroscopio, o sistema de imagen digital), para inspeccionar la lesión cutánea. Dicho instrumento, es capaz de amplificar la lesión en un rango de 6-100x [3], dependiendo del tipo de aparato usado. Igualmente, este aumento hace posible que se identifiquen varias estructuras de la superficie y subsuperficie de la lesión que no se pueden ver a simple vista (sin usar aparatos de aumento). En los últimos años, se ha producido un incremento considerable en el interés por desarrollar sistemas basados en el CAD (Computer Aided Diagnosis), con el fin de poder analizar imágenes dermatoscópicas, como las que se muestran en la Figura 1-1 [1]. El empleo de procesos que conlleven el uso de las técnicas anteriores (por dermatólogos) apoyadas en la dermatoscopia, favorece el incremento de la sensibilidad de la detección del melanoma entre un 10% y un 27% [3]. Por tanto, el principal objetivo de dichos sistemas es, sin duda, el diagnóstico del cáncer de piel, en particular del melanoma maligno [1], [4]. No obstante, se ha demostrado que la dermatoscopia incrementa el rendimiento en la diagnosis de melanomas, solo si los dermatólogos han recibido un entrenamiento formal [3].

También cabe destacar que a lo largo del diagnóstico de la lesión pigmentada, los dermatólogos distinguen hasta seis clases porcentuales no muy bien definidas, identificadas como colores: negro, marrón oscuro, marrón claro, azul-gris, rojo y blanco [4]. Asimismo, es sabido que las lesiones malignas presentan con mucha frecuencia más de tres colores, mientras que en lesiones benignas se suelen observar tres o menos, aunque es cierto que pigmentos como el negro, rojo, blanco y azul-gris se encuentran más asiduamente en melanomas [4].

Como ya ha sido mencionado en [1], la mayoría de las técnicas que se han descrito hasta la fecha, utilizan aproximaciones con patrones comunes de reconocimiento para el diagnóstico de las lesiones. Para ello, se extraen un gran número de características de bajo nivel que caracterizan la forma, el color e incluso la textura de la lesión y además se utilizan para el entrenamiento de clasificadores de imágenes [1]. Es importante destacar que este tipo de sistema basado en el diagnóstico por ordenador (CAD), no imita el análisis que lleva a cabo un dermatólogo, por lo que para la comunidad médica supone un avance difícil de aceptar. Esto se debe a que, para los especialistas dentro del campo médico, los métodos que sí intentan reproducir los procedimientos ya establecidos y validados por su comunidad, son más fiables que aquellos que implementan otros procedimientos de análisis y evaluación [1]. Una forma de poder lidiar con este problema, viene dada por una tendencia relativamente nueva en los sistemas CAD, que se fundamenta en el desarrollo de diferentes bloques de

procesamiento que imiten los algoritmos médicos de diagnóstico más utilizados [1]. Algunos de estos procesos de diagnóstico son: la regla ABCD [5], el método de Menzies [6], y la lista de comprobación de 7 puntos [7].

En este trabajo, se propone un algoritmo encargado de la detección de colores en imágenes dermatoscópicas, que como se describió con anterioridad, constituye una herramienta muy considerada por los dermatólogos a la hora de realizar su propio diagnóstico para este tipo de imágenes [1]. La metodología que se va a llevar a cabo en este documento, se basa principalmente en el cálculo de un modelo estadístico de 6 colores (negro, marrón oscuro, marrón claro, azul-gris, rojo y blanco), usando mezclas Gaussianas [1]. Con objeto de poder obtener dichos modelos, se emplea un conjunto de imágenes segmentadas proporcionadas por el especialista, que se usan para el entrenamiento del algoritmo aplicado [1]. Una vez que se obtienen las mezclas de cada color, estas se usan para determinar la aparición de pigmentos en conjuntos más grandes de imágenes, y más tarde se cuantifica el número de colores que hay en cada una de esas imágenes [1]. Previamente a lo mencionado con anterioridad, cabe destacar que las imágenes son debidamente corregidas antes de su posterior análisis y evaluación, eliminando pelos, burbujas o cualquier imperfección que pueda afectar a la diagnosis, además de aplicar otro tipo de correcciones que tienen que ver con la corrección gamma ($\gamma = 2.2$) [1] o los espacios HSV o CIE La^*b^* .

La estructura que sigue el documento es la siguiente: El capítulo 2 presenta el estado del arte, describiendo todos los algoritmos que se llevaron a cabo antes del que se expone en este trabajo, seguidamente, el capítulo 3 presenta la metodología, mientras que el capítulo 4 expone los resultados obtenidos y describe la base de datos empleada. Por último el capítulo 5 recoge las conclusiones obtenidas a lo largo del documento, mientras que el capítulo 6 expone nuevas soluciones y métodos futuros.

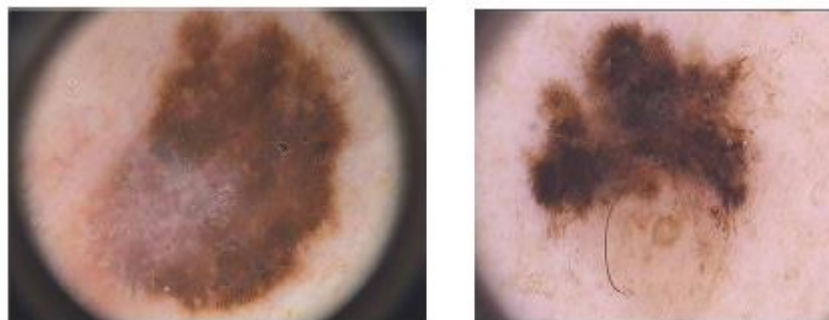


Figura 1-1. Ejemplos de imágenes dermatoscópicas.

2 ESTADO DEL ARTE

Es sabido que los enfoques que CCV(Classical Computer Vision) ha aportado, llevan consigo el problema de la clasificación de las lesiones de la piel a través de la extracción de características visuales de bajo nivel, así como de la presentación de las mismas como entradas para un clasificador[4]. Asimismo, entre estos enfoques hay algunos textos que se basan solamente en las características de color, mientras que otros emplean la información que se obtiene de ellas conjuntamente con otro tipo de características, tales como la textura o la forma [4]. Por otro lado, abandonando los puntos de vista clásicos y siguiendo una vertiente diferente, se observa que en trabajos mucho más recientes, las lesiones de la piel son clasificadas mediante redes neuronales convolucionales profundas [4]. No obstante, se puede destacar la existencia de determinados documentos que se centran esencialmente en detectar los seis colores principales que pueden presentar las lesiones cutáneas y que han sido nombrados en la introducción de este trabajo [4]. En primer lugar, Silva et al [8], enfocaron sus estudios en la posibilidad de separación de las seis posibles clases de colores, y llegaron a la conclusión de que algunas de ellas podían ser distinguidas en las imágenes RGB dermatoscópicas originales [4]. Seguidamente, como se muestra en [9], estos mismos autores emplearon varias redes neuronales para etiquetar cada pixel y clasificarlo dentro de las clases de colores posibles [4]. Por otra parte, Seidenari et al. [10], enfrentándose al reto de automatizar la implementación de la detección de pigmentos en imágenes dermatoscópicas, considerada hasta la fecha un proceso bastante subjetivo, presentaron un programa de ordenador que se encargaba de la evaluación de los colores presentes en los melanomas. Estos autores, para dicho propósito, se encargaron del desarrollo de una paleta de colores formada por un total de 98 parches de color, siendo 15 de ellos negros, 10 marrón oscuro, 28 marrón claro, 9 rojos, 12 blancos y 24 azules [4]. A continuación, cada uno de los píxeles de una lesión nueva era asignado al parche de color que minimizara la distancia Euclídea en el espacio de color RGB, y más tarde los mismo autores [11], junto con Zortea et al. [12] se hicieron cargo de la extracción y obtención de las características que se apoyaban en este enfoque, el cual se fundamentaba en la creación de una paleta de color [7]. Dichas características eran necesarias para distinguir las lesiones benignas de las malignas [4].

Otros autores que siguieron esta vertiente, fueron Barata et al. [1] que expusieron la detección automática de cinco colores (negro, marrón oscuro, marrón claro, azul-gris y blanco) en vez de los seis que se dan usualmente en las imágenes dermatoscópicas. El proceso que seguían estos autores se basaba en primera instancia, en el aprendizaje de un modelo de mezcla Gaussiano para cada uno de los cinco colores y seguidamente, se procedía al muestreo de cada lesión en parches cuadrados calculándose las probabilidades de pertenecer a cada una de las mezclas[1],[4].

Es importante subrayar que, como se observa en [4], el análisis de colores en las imágenes dermatoscópicas también se aplica a la detección de líneas específicas de color. Stoecker et al. [13] presentaron la idea de la detección de lesiones cutáneas mediante el empleo de umbrales en los planos de color tanto rojo como verde [4]. De igual manera, Madasu y Lovell [14], detectaron manchas, pero de una forma bastante distinta a la de [13] basada en la aplicación de una técnica muy confusa de co-agrupamiento para la segmentación del color. Varios autores como Celebi et al [15] en cambio, se centraron en la detección de especies de velos azules blanquecinos, mediante el cálculo absoluto y relativo de las características del color, clasificándolas a través de un árbol de decisión [4]. En la misma línea que [17], Di Leo et al., propusieron, en [18], un sistema que también se centraba en la estimación de velos azulados blanquecinos, pero que a diferencia del anterior método, empleaban el procesador de imágenes ELM (Epiluminescence Microscopy), con el fin de analizar de forma automática las lesiones pigmentadas.

A su vez, De Vita et al [17], se encargan en su trabajo, del cálculo de la media y de la varianza en tres espacios de colores diferentes y aplican un modelo de árbol lógico [4].

3 METODOLOGÍA

En este capítulo se explicará la metodología seguida para el aprendizaje de las seis mezclas Gaussianas obtenidas, con el fin de representar cada uno de los colores, así como el método usado para identificar dichos colores en una lesión de la piel y cuantificarlos. La técnica que aquí se lleva a cabo consta de dos partes, como bien se ha podido intuir. La primera de ellas, se encarga del aprendizaje de las mezclas de color Gaussianas, mientras que la segunda realiza la asignación y cuantificación de los colores presentes en las lesiones proporcionadas. A continuación, a parte de describir ambas partes del método descrito, se realizará un esquema de ese método, con el fin de hacer mucho más clara su explicación y comprensión. Este esquema se muestra en la Figuras 3-1y 3-2. Mencionar además que los códigos de las funciones de Matlab que en los esquemas se muestran, se encuentran en el Anexo A: Códigos.

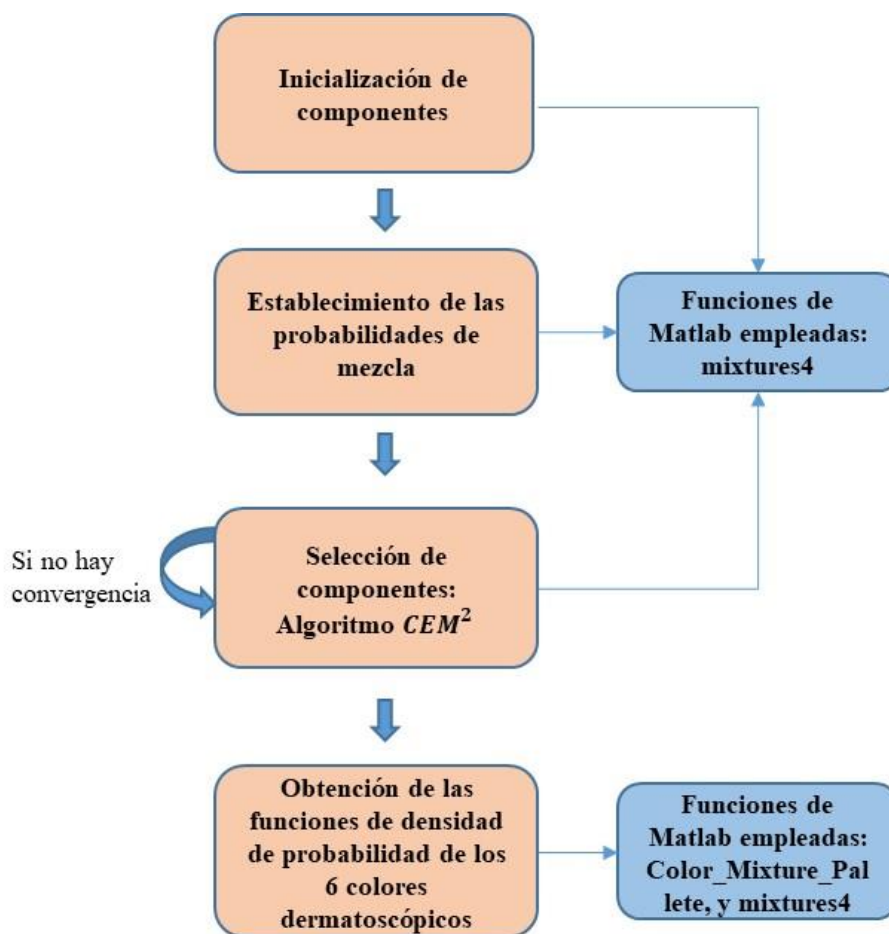


Figura 3-1. Esquema de la primera parte del método propuesto: Aprendizaje de modelos de mezcla Gaussiana

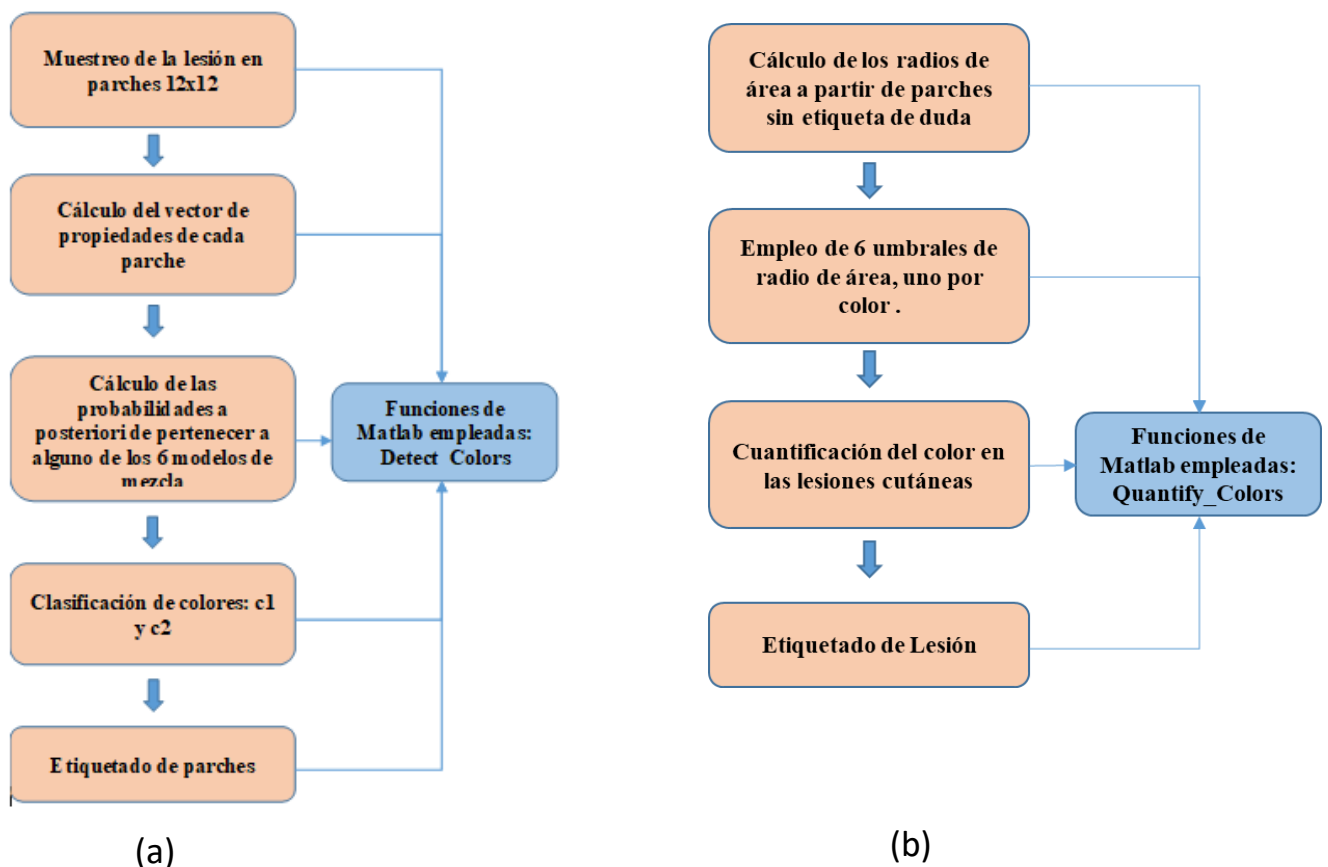


Figura 3-2. Esquema de la segunda parte del método propuesto: Asignación (a) y cuantificación (b) de colores

3.1 Modelo de aprendizaje de la mezcla de color

La técnica más común utilizada para ajustar los modelos finitos de mezcla a los datos obtenidos, es el algoritmo EM (Expectation Maximization), propuesto en [18], el cual tiende a una estimación de ML (Maximun Likelihood) o MAP (Maximun a Posteriori) de los parámetros de la mezcla, y concreto, para las mezclas Gaussianas, la convergencia de dicho algoritmo se estudia exhaustivamente en [19] y [20]. Pero este método, para el caso de los modelos finitos de mezclas, los cuales se explicarán posteriormente, presenta una serie de desventajas:

- Es un método local, es decir toma decisiones en función de la información que obtienen en el mismo momento, por lo que es depende fuertemente del proceso de inicialización llevado a cabo [18].
- Es posible que converja en el límite del espacio de parámetros, donde la probabilidad es ilimitada, lo que lleva a la obtención de estimaciones que no aportan ningún tipo de información [18].

De esta manera, se puede concluir que usando el mismo, es muy difícil obtener una selección adecuada del mejor número de componentes de la mezcla. Por tanto, para tratar de solventar este problema, se presenta en [18] una alternativa que se apoya en el empleo de una variante del mismo algoritmo EM, llamada MML, que implementa el criterio de la mínima longitud del mensaje como la función de coste, con el fin de encontrar de forma automática el mejor número de componentes de una mezcla [1]. Este criterio, además de todo lo señalado, es menos sensible a la inicialización que EM, y por supuesto, evita el problema del límite del espacio de parámetros [18].

Sin embargo, en este trabajo se propone un algoritmo no supervisado para el aprendizaje de estos modelos finitos de mezcla a partir de gran cantidad de datos. Tal y como se explica en [18], este método se caracteriza por ser no supervisado ya que es capaz de hacer una selección automática de los componentes que forman las mezclas, y además a diferencia del método EM brevemente mencionado, no es dependiente del proceso de inicialización en el sentido de que no necesita realizar una inicialización cuidadosa. A parte de evitar estos dos inconvenientes que presenta el algoritmo EM, la técnica propuesta consigue eludir también el problema de convergencia en el límite del espacio de parámetros.

No obstante, a parte de mejorar ciertos aspectos del algoritmo EM, el método planteado presenta una novedad frente al mismo. Esta técnica integra en un solo algoritmo tanto los procesos de estimación como los de selección de modelos, es decir no utiliza un criterio de selección de modelo a parte con el que se seleccionaría un modelo candidato preestimado [18]. En conclusión se puede decir en este trabajo se presenta un criterio de inferencia y un algoritmo encargado de su implementación.

Antes de analizar y describir el funcionamiento de dicho algoritmo, se hará un breve desarrollo de los modelos finitos de mezclas con el fin de poder entender mejor este proceso.

3.1.1 Modelos finitos de mezclas

Como se propone en [1] y [18], en este trabajo se usan modelos estadísticos o finitos para aprender colores, sacados a partir de mezclas Gaussianas. Estos pigmentos se aprenden de forma separada a través de modelos de mezcla Gaussianas, consiguiendo así poder definir, mediante las mismas, los 6 colores dermatoscópicos (negro, azul-gris, marrón oscuro, marrón claro, rojo y blanco). Por tanto, la paleta de colores que se obtiene finalmente recoge seis mezclas gaussianas diferentes, teniendo una función de densidad de probabilidad [1] por mezcla, como la que se presenta a continuación:

$$p(\mathbf{y}|c, \theta^c) = \sum_{m=1}^{k_c} \alpha_m^c p(\mathbf{y}|c, \theta_m^c) \quad (3-1)$$

En la ecuación (3-1) se observan elementos como, $c = 1, 2, \dots, 6$ que se refiere a cada uno de los colores dermatoscópicos o k_c que representa el número de componentes que forman la mezcla de color. De igual manera, cada θ_m^c refleja el conjunto de parámetros que definen el m-ésimo componente de la mezcla, $\theta^c = \{\theta_1^c, \dots, \theta_k^c, \alpha_1^c, \dots, \alpha_k^c\}$ es el conjunto completo de parámetros necesarios para especificar la mezcla, [1], [18]. Por otra parte, $\alpha_1^c \dots \alpha_k^c$ son las probabilidades de mezcla, que por supuesto deben satisfacer las siguientes condiciones [18]:

$$\alpha_m^c \geq 0, m = 1, \dots, k_c \text{ y } \sum_{m=1}^{k_c} \alpha_m^c = 1 \quad (3-2)$$

Además, en este trabajo se establece $\mathbf{Y} = [Y_1, \dots, Y_d]^T$ como una variable aleatoria de dimensión d , que sigue una distribución de mezcla finita de k_c componentes siempre y cuando su función de densidad de probabilidad se corresponda con la de (3-1), siendo $\mathbf{y} = [y_1, \dots, y_d]^T$ un vector característico que representa en particular una salida de \mathbf{Y} [18] y está compuesto por los colores observados, además de utilizarse en la siguiente expresión [1],

$$p(\mathbf{y}|c, \theta_m^c) = \frac{(2\pi)^{-\frac{d}{2}}}{\sqrt{|R_m^c|}} \exp\left\{-\frac{1}{2}(\mathbf{y} - \mu_m^c)^T R_m^{-1}(\mathbf{y} - \mu_m^c)\right\} \quad (3-3)$$

donde $\theta_m^c = (\mu_m^c, R_m^c, \alpha_m^c)$ caracteriza a la gran parte de componentes del sistema [18]. Así pues, se puede concluir que los parámetros que se van a estimar cuando el algoritmo aprende una mezcla, son claramente la media y la covarianza de cada componente (μ_m^c, R_m^c) , así como las correspondientes probabilidades de mezcla α_m^c [1].

Continuando con la descripción de este tipo de modelos, se puede decir que dado un conjunto de n muestras independientes e igualmente distribuidas $Y = \{\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n)}\}$ la probabilidad logarítmica correspondiente a una mezcla de componente k_c es [18]

$$\log p(Y|\theta^c) = \log \prod_{i=1}^n p(\mathbf{y}^{(i)}|\theta^c) = \sum_{i=1}^n \log \sum_{m=1}^{k_c} \alpha_m^c p(\mathbf{y}^{(i)}|c, \theta_m^c) \quad (3-4)$$

De la misma manera, se puede observar que la estimación ML

$$\hat{\theta}_{ML}^c = \arg \max_{\theta} \{\log p(Y|\theta^c)\} \quad (3-5)$$

no puede ser hallada de forma analítica, al igual que el criterio MAP Bayesiano que se muestra a continuación. Ambos criterios o estimaciones, ML y MAP, dependen de las restricciones reflejadas en (3-2) [18].

$$\hat{\theta}_{MAP}^c = \arg \max_{\theta} \{\log p(Y|\theta^c) + \log p(\theta^c)\} \quad (3-6)$$

3.1.2 Criterio propuesto

3.1.2.1 Introducción al criterio propuesto

Como se comprueba en [18], los métodos deterministas detallados en ese mismo artículo, son criterios de selección de clase de modelo. Esto significa que se realiza la selección de la clase de modelo, denominada M_{k_c} , según su mejor representante, es decir $\hat{\theta}(k_c)$. Sin embargo, la distinción que se hace en los modelos de mezclas entre la selección de la clase de modelo y la estimación del modelo, no tiene mucho valor. Una vez concluido esto, se puede observar la situación desde una perspectiva distinta: se presenta k_c como un gran valor arbitrario que infiere en la estructura de la mezcla, dejando que las estimaciones de algunas probabilidades de mezcla sean cero. Este nuevo enfoque, coincide con los principios en los que se basa MML, el cual no ha adoptado la jerarquía clase de modelo, si no que persigue encontrar de forma directa el modelo óptimo entre todos los modelos disponibles en el conjunto

$$\bigcup_{k_c=k_{c,min}}^{k_{c,max}} M_{k_c} \quad (3-7)$$

en vez de seleccionar uno entre un conjunto de modelos candidatos $\{\hat{\theta}(k_c), k_c = k_{c,min}, \dots, k_c = k_{c,max}\}$. También es sabido que otros usos de MML para mezclas, no han seguido exactamente esta nueva visión, sino que han terminado usando MML como un criterio de selección de clase de modelo [24].

3.1.2.2 Descripción del criterio MML en términos generales

Para comenzar este apartado, decir que la idea detrás de los criterios que se apoyan en la mínima longitud de codificación, como pueden ser MDL y el propio MML, es aquella que expone que, si se es capaz de construir un pequeño código para unos datos determinados, entonces se tendrá un buen modelo de generación de información [18].

Para entender esta idea desde un punto de vista más formal, se define Y como un conjunto de datos, producido de acuerdo a $p(Y|\theta^c)$, el cual ha de ser codificado y transmitido [18]. Según Shannon [25], la longitud de código más corta (esta se mide en bits si se usa logaritmo en base 2, o en nats si se usa logaritmo natural) para el conjunto Y es $[-\log p(Y|\theta^c)]$, donde $[a]$ denota “el entero más pequeño, pero no menor que a ” [18]. Como incluso para grandes conjunto de datos $-\log p(Y|\theta^c) \gg 1$, el operador $[.]$ normalmente se quita [18].

Si $p(Y|\theta^c)$ es conocida tanto en el transmisor como en el receptor, ambos tendrán el mismo código y la comunicación se llevará a cabo, mientras que, si θ^c es desconocido en primera instancia, el transmisor tendrá que empezar estimando y enviando θ^c . Esta situación hace que el mensaje se divida en dos, cuya longitud total es dada por la siguiente ecuación:

$$\text{Length}(\theta^c, Y) = \text{Length}(\theta^c) + \text{Length}(Y|\theta^c) \quad (3-8)$$

Usualmente, casi todos los criterios que se apoyan en la mínima longitud de codificación, como son por ejemplo MDL y MML, afirman que el parámetro estimado es el que minimiza $\text{Length}(\theta^c, Y)$ [18]. El aspecto fundamental de esta idea reside en el parámetro θ^c . La cuestión es que ya que θ^c es un vector cuyas componentes son reales, una longitud de código de carácter finito solo puede ser obtenida cuantizando θ^c a precisión finita. Lo que se explica a continuación. En primer lugar, se considera $\tilde{\theta}^c$ como la versión cuantizada de θ^c , seguidamente si se usa una precisión detallada se obtiene que $\text{Length}(\tilde{\theta}^c)$ es bastante grande, mientras que en cambio, $\text{Length}(Y|\tilde{\theta}^c)$ se hace muchísimo más pequeña debido a que $\tilde{\theta}^c$ se acerca mucho al valor óptimo. Sin embargo, si en vez de una precisión detallada, usamos una precisión más ordinaria y de calidad inferior, el resultado obtenido indica que $\text{Length}(\tilde{\theta}^c)$ se hace pequeña, mientras que a diferencia de lo que pasaba en la situación anterior, $\text{Length}(Y|\tilde{\theta}^c)$ se aleja muchísimo del valor óptimo. Con esto se concluye que hay muchísimas formas de formalizar esta idea [18].

Por otra parte, se ha de resaltar que no supone ningún obstáculo el hecho de que la información tenga valor real, simplemente hay que trincar Y con una precisión detallada arbitraria denominada δ , además de reemplazar la densidad $p(Y|\theta^c)$ por la probabilidad $p(Y|\theta^c)\delta^d$ (siendo d la dimensión de Y). La longitud de código que se obtiene finalmente es $-\log p(Y|\theta^c) - d \log \delta$, siendo el último miembro una constante aditiva irrelevante.

La forma característica del planteamiento de MML que aquí se adopta, lleva a proponer un criterio en el que la minimización con respecto θ^c es simultánea tanto en θ^c como en C , donde C representa la dimensión de θ^c en este contexto [18]. El criterio propuesto se presenta a continuación:

$$\hat{\theta}^c = \arg \min_{\theta^c} \left\{ -\log p(\theta^c) - \log p(Y|\theta^c) + \frac{1}{2} \log |\mathbf{I}(\theta^c)| + \frac{C}{2} \left(1 + \log \frac{1}{12} \right) \right\} \quad (3-9)$$

Los términos que componen el anterior criterio son definidos a continuación. El primer término que aparece es $\mathbf{I}(\theta^c) \equiv -E[D_{\theta^c}^2 \log p(Y|\theta^c)]$, y representa la matriz de información de Fisher esperada, mientras que $|\mathbf{I}(\theta^c)|$ es el determinante de dicha matriz. Ha de comentarse que $D_{\theta^c}^2$ representa la matriz de segundas derivadas o hessiana.

Por otra parte el criterio MDL, el cual no coincide de manera conceptual pero si formal con el BIC (Bayesian Information Criterion), se puede obtener como una aproximación de (3-9). Primeramente se empieza asumiendo lo que se llama en inglés una “flat prior” a la que se denomina en este trabajo como $p(\theta^c)$, después de realizar esa asunción se retira. Una vez completado este paso, se concluye que como $\mathbf{I}(\theta^c) = n \mathbf{I}^{(1)}(\theta^c)$ entonces $\log |\mathbf{I}(\theta^c)| = C \log n + \log |\mathbf{I}^{(1)}(\theta^c)|$, siendo $\mathbf{I}^{(1)}(\theta^c)$ la información de Fisher correspondiente a una única

observación. Para grandes valores de n se retiran o eliminan los términos de orden 1 tales como $\log |\mathbf{I}^{(1)}(\boldsymbol{\theta}^c)|$ y $\frac{C}{2} \left(1 + \log \frac{1}{12}\right)$. Después de ello y como paso final, se toma, para una C dada, $-\log(p(Y|\boldsymbol{\theta}^c)) \cong -\log(p(Y|\widehat{\boldsymbol{\theta}}^c(C)))$, donde se observa que el término $\widehat{\boldsymbol{\theta}}^c(C)$ es la estimación ML correspondiente [18]. El resultado de todo esto es el criterio MDL

$$\hat{C}_{\text{MDL}} = \arg \min_C \left\{ -\log(p(Y|\widehat{\boldsymbol{\theta}}^c(C))) + \frac{C}{2} \log n \right\} \quad (3-10)$$

donde la longitud del código de datos es $-\log(p(Y|\widehat{\boldsymbol{\theta}}^c(C)))$, mientras que la longitud de código de cada uno de los componentes C que forman parte de $\widehat{\boldsymbol{\theta}}^c(C)$, tiene que ser proporcional a $\frac{1}{2} \log n$. Esta diferencia entre esas dos partes indica que la precisión de codificación que presentan las estimaciones de los parámetros, se lleva a cabo de manera inversamente proporcional a la desviación estándar del error de estimación. Dicho error, normalmente reduce su valor con la \sqrt{n} cuando es sometido a ciertas condiciones de regularidad, así este decremento conduce al término $\frac{1}{2} \log n$ [18].

3.1.2.3 Desarrollo y funcionamiento del criterio propuesto para mezclas

Una vez visto de forma general el criterio MML, se va a pasar a la descripción del funcionamiento del criterio propuesto, pero en este caso, al utilizar mezclas. Para ello, se redefinirán funcionalidades atribuidas a parámetros anteriormente descritos, además de introducir nuevos factores para un mejor entendimiento del criterio.

Para empezar, lo primero que hay que destacar es que para las mezclas, la componente $\mathbf{I}(\boldsymbol{\theta}^c)$ usualmente no se puede obtener de manera analítica [18]. Para solventar este problema, tal y como se describe en el [18], se va a reemplazar dicha componente $\mathbf{I}(\boldsymbol{\theta}^c)$ por la matriz completa de información de Fisher, es decir $\mathbf{I}_C(\boldsymbol{\theta}^c) \equiv -E[D_{\boldsymbol{\theta}^c}^2 \log p(Y, Z|\boldsymbol{\theta}^c)]$, cuyos límites superiores se corresponden con $\mathbf{I}(\boldsymbol{\theta}^c)$. Esta matriz completa $\mathbf{I}_C(\boldsymbol{\theta}^c)$, presenta una estructura en forma de bloque diagonal, esto quiere decir que la matriz $\mathbf{I}_C(\boldsymbol{\theta}^c)$ es una matriz diagonal cuadrada, donde los elementos que componen su diagonal son matrices cuadradas de cualquier tamaño disponible. Los elementos que no forman parte de la diagonal son, por supuesto, ceros. A continuación se ilustra lo descrito, mediante la siguiente ecuación

$$\mathbf{I}_C(\boldsymbol{\theta}^c) = n \text{ block-diag} \{ \alpha_1^c \mathbf{I}^{(1)}(\boldsymbol{\theta}_1^c), \dots, \alpha_{k_c}^c \mathbf{I}^{(1)}(\boldsymbol{\theta}_{k_c}^c), \mathbf{M} \} \quad (3-11)$$

donde $\mathbf{I}^{(1)}(\boldsymbol{\theta}_m^c)$ es la matriz de Fisher para una observación única que se ha producido por el componente m -ésimo, y \mathbf{M} es la matriz de Fisher de una distribución multinomial, siendo además $|\mathbf{M}| = (\alpha_1^c \alpha_1^c \cdots \alpha_{k_c}^c)^{-1}$. Esta aproximación de $\mathbf{I}(\boldsymbol{\theta}^c)$ a través de $\mathbf{I}_C(\boldsymbol{\theta}^c)$ se vuelve exacta justo en el límite de los componentes que no se superponen [18]. Seguidamente se adoptan una serie de expresiones, que a priori, contienen una falta de conocimiento de los parámetros de la mezcla. Por supuesto, se modelan los parámetros de los diferentes componentes como si fueran independientes también a priori, a su vez siendo independientes de las probabilidades de mezcla, como se muestra a continuación

$$p(\boldsymbol{\theta}^c) = p(\alpha_1^c, \dots, \alpha_{k_c}^c) \prod_{m=1}^{k_c} p(\boldsymbol{\theta}_m^c) \quad (3-12)$$

Para cada uno de los componentes de la expresión anterior se adopta el estándar no informativo de Jeffreys

$$p(\boldsymbol{\theta}_m^c) \propto \sqrt{|\mathbf{I}^{(1)}(\boldsymbol{\theta}_m^c)|} \quad (3-13)$$

$$p(\alpha_1^c, \dots, \alpha_{k_c}^c) \propto \sqrt{|M|} = (\alpha_1^c \alpha_2^c \dots \alpha_{k_c}^c)^{-1/2}$$

expresiones que se cumplen si $0 \leq \alpha_1^c, \alpha_2^c, \dots, \alpha_{k_c}^c \leq 1$ y $\alpha_1^c + \alpha_2^c + \dots + \alpha_{k_c}^c = 1$ [18]. Una vez realizadas estas elecciones y sabiendo que para una mezcla de k_c componentes el parámetro C presenta el siguiente valor $C = Nk_c + k_c$ (siendo N el número de parámetros que designan a cada componente), se comprueba que la ecuación número (3-9) se transforma en la siguiente

$$\hat{\theta}^c = \arg \min_{\theta^c} L(\theta^c, Y) \quad (3-14)$$

con

$$L(\theta^c, Y) = \frac{N}{2} \sum_{m=1}^{k_c} \log\left(\frac{n\alpha_m^c}{12}\right) + \frac{k_c}{2} \log \frac{n}{2} + \frac{k_c(N+1)}{2} - \log p(Y|\theta^c) \quad (3-15)$$

Además del término de orden $1 \frac{k_c(N+1)}{2} (1 - \log 12)$, el criterio propuesto presenta una interesante e intuitiva interpretación del estándar de la formulación del código de dos partes característico de MDL y MML:

1. Se establece $-\log(p(Y|\theta^c))$ como la longitud de código de los datos
2. Se designa $n\alpha_m^c$ como el número de puntos esperados de datos generados por la componente m -ésima de la mezcla. Dicha designación, se puede considerar un tamaño de muestra eficiente a partir de la cual se puede estimar θ_m^c , por tanto la longitud de código óptima para cada θ_m^c , desde el punto de vista del criterio MDL, es $(N/2) \log(n\alpha_m^c)$.
3. Los factores α_m^c se estiman a partir de todas las n observaciones, lo que da pie al término $\frac{k_c}{2} \log(n)$.

Es importante resaltar que si los factores que se mostraban en el punto 3 de la lista anterior fueran cero, entonces la función objetiva en (3-15) no tiene ningún sentido ya que tiende a convertirse en $-\infty$. Este problema se puede eliminar siempre y cuando se siga su interpretación de la longitud de código. Para hacer que el modelo de mezcla sea más específico, solo se necesita que los parámetros de las componentes cuya probabilidad no es cero sean codificados. Con $k_{c,nz}$ denotando la cantidad de componentes que tienen una probabilidad distinta de cero, obtenemos la siguiente expresión

$$L(\theta^c, Y) = \frac{N}{2} \sum_{m: \alpha_m^c > 0} \log\left(\frac{n\alpha_m^c}{12}\right) + \frac{k_{c,nz}}{2} \log \frac{n}{2} + \frac{k_{c,nz}(N+1)}{2} - \log p(Y|\theta^c) \quad (3-16)$$

Aunque se necesita un término adicional para poder realizar la codificación de $k_{c,nz}$, se sabe que su longitud de código es constante (especialmente, $\log(k_c)$ ya que $k_{c,nz} \in \{1, 2, \dots, k_c\}$), por lo que dicho término es irrelevante. Por tanto la expresión (3-16) se consolida como la función de coste cuya minimización respecto de θ^c constituirá la estimación de la mezcla deseada en este trabajo [18].

3.1.3 Algoritmo utilizado para la implementación del criterio propuesto

Como se explica en el [1], estos algoritmos de minimización ponen a prueba todas las componentes desde $\{k_{c,min}, k_{c,min} + 1, \dots, k_{c,max} - 1, k_{c,max}\}$ usando CEM (Component Wise Expectation Maximization) para las mezclas. Seguidamente, una vez que se consigue la convergencia con una k_c determinada, se pone a cero el componente con menor $\hat{\alpha}_m^c$ y se vuelve a realizar CEM hasta llegar de nuevo a la convergencia. Este procedimiento se hace siempre y cuando se cumpla que $k_c \geq k_{c,min}$, siendo de esta forma los parámetros estimados, así como el número de componentes, los encargados de minimizar el criterio MML. A continuación

se describirá con mucho más detalle lo anteriormente explicado.

3.1.3.1 Minimización de la función de coste mediante EM

Si se analiza la expresión anterior, es decir la (3-16), desde un enfoque Bayesiano, se puede observar que ésta es equivalente, siempre y cuando $k_{c,nz}$ sea fija, a una densidad a posteriori que resulta de la adopción del DTP(Dirichlet-Type Prior) para los factores α_m^c

$$p(\alpha_1^c, \dots, \alpha_{k_c}^c) \propto \exp \left\{ -\frac{N}{2} \sum_{m=1}^{k_c} \log \alpha_m^c \right\} \quad (3-17)$$

y de una “flat prior” que conduce a estimaciones del tipo ML para las θ_m^c . Por otra parte, como los DTPs son conjugados como probabilidades multinomiales [18], el algoritmo EM hace uso del Paso M (3-2) para conseguir la minimización de la función de coste presentada en la ecuación (3-16), con $k_{c,nz}$ fija:

$$\widehat{\alpha}_m^c(t+1) = \frac{\max \left\{ 0, \left(\sum_{i=1}^n w_m^{(i)} \right) - \frac{N}{2} \right\}}{\sum_{j=1}^{k_c} \max \left\{ 0, \left(\sum_{i=1}^n w_j^{(i)} \right) - \frac{N}{2} \right\}} \quad (3-18)$$

para $m = 1, 2, \dots, k_c$,

$$\widehat{\theta}_m^c(t+1) = \arg \max_{\theta_m^c} Q \left(\theta^c, \widehat{\theta}^c(t) \right), \text{ para } m: \widehat{\alpha}_m^c(t+1) > 0$$

donde las $w_m^{(i)}$ son proporcionadas por la ecuación del Paso E [18]. También se ha de resaltar que, como se venía observando en la ecuación (3-4), cualquier componente cuya $\alpha_m^c = 0$, no contribuye a la probabilidad logarítmica, y por tanto los parámetros θ_m^c que se corresponden con las componentes para las que $\widehat{\alpha}_m^c(t+1) = 0$, son irrelevantes.

Además de todo ello, se destaca que para $N > 1$, este DTP con exponentes negativos $-N/2$ no es el principio original de Jeffreys que se usa en las probabilidades de mezcla. Esta consideración se hace, como se menciona en [18], ya que, aunque los DTPs con exponentes negativos sí que se han empleado y asumido en gran cantidad de contextos, en [18] se considera que estos priors que poseen exponentes con valor por debajo de -1, y a los cuales se les da el atributo de impropios, no se han usado antes.

A continuación, se nombrarán algunas facetas importantes de este algoritmo.

- **Eliminación de componentes**

Observando la ecuación número (3-18), que representa una característica bastante interesante del Paso M [18], se puede resaltar que la misma realiza eliminación de componentes. Esta particularidad indica que la ecuación (3-18) se convierte en una regla explícita para pasar del valor del parámetro $k_{c,nz}$ que se tiene, a un valor más pequeño del mismo. Además, ha de acentuarse que esta propiedad presenta un claro beneficio para el algoritmo ya que previene que se acerque al límite del espacio de parámetro, eliminándose así uno de los problemas que presenta el estándar EM para mezclas [18].

- **Robustez con respecto a la inicialización**

Como se menciona en el [1], este algoritmo es muy robusto con respecto a la inicialización. Eso es debido a que se empieza inicializando la componente $k_{c,nz}$ con el siguiente valor $k_{c,nz} = k_c$, donde k_c es un valor mucho más grande que la cantidad c óptima de componentes de la mezcla. El hecho de que el algoritmo EM sea incapaz de mover componentes a través de regiones de baja probabilidad, provoca que el máximo local de la

probabilidad se eleve cuando hay muchas componentes en una región del espacio, y muy pocas en otra [18]. Si se empieza inicializando con muchas componentes repartidas por todo el espacio, además de eliminar aquellas que son innecesarias, el problema anteriormente planteado se evitaría. Por otra parte, también se destaca la existencia de otra clase de mínimo local, para la que su estándar EM igualmente produce problemas, y además se refiere a situaciones en las que dos o más componentes poseen factores parecidos, por lo que aproximadamente comparten los mismos puntos de datos. Esas situaciones que se han mencionado, presentan un carácter inestable debido al DTP con exponentes negativos, observado en la expresión (3-17). Este principio también provoca que los componentes compitan entre ellos haciendo que uno de ellos, finalmente, gane y los demás sean eliminados.

A parte de estas facetas, el algoritmo presentado está relacionado con el aprendizaje supervisado, así como con los principios de mínima entropía, comentados en [18], en el que además aparecen descritos con un mayor detalle.

3.1.3.2 El algoritmo CEM: Introducción

Como se ha recalcado durante el desarrollo de este capítulo, el uso de EM acarrea una serie de problemas. De una forma más concreta, se puede decir que el uso del algoritmo EM de manera directa con el Paso M, mostrado en las expresiones de (3-18), tiene un modo de fallo que ocurre cuando el parámetro k_c es demasiado grande. Esta situación problemática se puede eludir empleando el reciente algoritmo CEM para mezclas, es decir el CEM² como se ilustra en [18] y [26]. El funcionamiento del algoritmo CEM² se basa en la actualización de todos los parámetros del tipo α_m^c y θ_m^c de manera secuencial. Una de las características más importantes de CEM², es la que indica que si un componente es eliminado, es decir $\widehat{\alpha}_m^c(t+1) = 0$, la inmediata redistribución de su masa de probabilidad entre los demás componentes aumenta las oportunidades de sobrevivir de los mismos, permitiendo así la inicialización con una k_c arbitraria bastante grande sin presentar ningún tipo de problema. Otro aspecto a tener en cuenta, el cual se menciona tanto en [26] como en [18], es que el método de obtención de resultados, que se presenta en este capítulo mediante el algoritmo CEM², se realiza a través de la adopción de un procedimiento de actualización cíclico simple.

Por otra parte, aunque se pueda creer que CEM² es un algoritmo que, desde el punto de vista computacional, es mucho más pesado que el estándar EM, la realidad es que no es así. CEM² es un algoritmo que se considera ligeramente, más pesado que EM desde el punto de vista computacional.

3.1.3.3 Algoritmo completo

Una vez visto el procedimiento descrito en el subapartado superior, se puede resumir, tal y como viene indicado en [18], que después de conseguir la convergencia de CEM², se debe comprobar si existen valores más pequeños de $L(\theta^c, Y)$ cuando se ponen a cero las componentes que no fueron eliminadas por (3-18). Para ello, se propone en [18] que se eliminen las componentes que tengan el parámetro $\widehat{\alpha}_m^c$ más pequeño, es decir, sean las menos probables. Tras dicho proceso, se vuelve a aplicar el algoritmo CEM² hasta que vuelva a llegar a la situación de convergencia. Este procedimiento se repite hasta alcanzar $k_{c,nz} = 1$, escogiéndose la estimación que proporcione el valor mínimo de $L(\theta^c, Y)$ [18]. Sin embargo, si el número de componentes no consigue ser menor que un determinado parámetro, es decir, $k_{c,min} > 1$, entonces el procedimiento anterior se repetiría, pero esta vez hasta alcanzar dicho valor $k_{c,nz} = k_{c,min}$.

3.2 Asignación y cuantificación del color

En esta sección se verá el proceso llevado a cabo para la identificación de colores en una lesión de la piel, tal y como se explica en el [1]. Para ello, se adopta un esquema jerárquico en base al cual se tomarán decisiones, y en el que se dan principalmente dos pasos: Etiquetado de parche y etiquetado de lesión [1].

A continuación, se hará una descripción más detallada de los procedimientos de asignación y cuantificación del color, basados en el esquema anterior.

3.2.1 Procedimiento para la asignación del color

De acuerdo a lo enunciado en [1], el proceso para la asignación de colores, tiene su inicio en el muestreo de la lesión en pequeños parches de tamaño 12 x 12, usando una cuadrícula regular. Una vez realizado ese muestreo, se calcula un vector de propiedades encargado de caracterizar cada parche obtenido previamente. Para caracterizar dichos parches se usa el llamado color medio.

Con el fin de poder asignar una etiqueta con un determinado color a cada uno de los parches obtenidos, se calculan sus probabilidades a posteriori de pertenecer a cada uno de los modelos de colores conseguidos en los puntos precedentes [1]. Dichas probabilidades vienen dadas por la expresión siguiente,

$$p(c|\mathbf{y}) = \frac{p(\mathbf{y}|c, \hat{\theta}^c)p(c)}{p(\mathbf{y}|\hat{\theta})} \quad (3-19)$$

donde $\hat{\theta}^c = (\hat{\mu}^c, \hat{R}^c, \hat{\alpha}^c)$ ($\hat{\mu}^c, \hat{R}^c$ se trata de la media y covarianza), $\hat{\theta} = (\hat{\theta}^1, \dots, \hat{\theta}^6)$, y por último $p(c) = 1/6$, configurado para ser igual para todos los colores [1]. Al conseguir las probabilidades a posteriori, se clasifican los colores según su grado de membresía en los parches analizados y posteriormente, se eligen los colores que han obtenido el valor más alto y el segundo más alto, denotándose como c_1 y c_2 respectivamente.

Finalmente, se da una de las siguientes acciones:

1. Si $p(c_1|\mathbf{y}) \geq \delta$ y $p(c_1|\mathbf{y}) - p(c_2|\mathbf{y}) > \epsilon$, donde δ y ϵ son umbrales determinados empíricamente, el parche se etiqueta de acuerdo al color c_1 [1]
2. Si $p(c_1|\mathbf{y}) \geq \delta$ y $p(c_1|\mathbf{y}) - p(c_2|\mathbf{y}) \leq \epsilon$, el parche se marca una etiqueta que indica que hay duda entre ambos colores c_1 y c_2 [1].
3. Si $p(c_1|\mathbf{y}) < \delta$, el parche se rechaza[1]

En Matlab se hace esta asignación, con el Código 2 del Anexo A: Códigos.

3.2.2 Procedimiento para la cuantificación del color

En esta subsección, se explicará el procedimiento para la cuantificación del color, que es el paso a seguir después de la asignación. Este proceso consiste, fundamentalmente, en decidir si un color está presente o no en una lesión de la piel (Etiquetado de Lesión) [1]. Para ello, se utilizan seis umbrales de radio de área determinados empíricamente. Esto indica claramente que, cada región de color, solamente va a ser considerada si su radio de área está por encima de un umbral específico [1]. Los umbrales definidos son : un 15% del área de la lesión para el marrón claro, 5% para el marrón oscuro, 4% para el azul-gris, 2% tanto para el blanco como para el negro, y 3% para el rojo. Por supuesto, no hace falta decir que para calcular los radios de área solamente se usan los parches que no se han marcado con la etiqueta de duda.

En Matlab se hace esta clasificación, con el Código 3 del Anexo A: Códigos.

4 RESULTADOS

En este capítulo se presentan los resultados experimentales obtenidos a partir de la aplicación del algoritmo descrito en el capítulo 3, el cual se ha implementado en lenguaje Matlab recogido en el Anexo A: Códigos del trabajo. Muchos de los resultados expuestos se han obtenido apartir de las Tablas recogidas en el Anexo B: Datos Relevantes.

4.1 Base de datos de imagen

La base de datos de imagen, presentada en este trabajo, está compuesta por un total de 217 imágenes de tamaño 768x512 en formato RGB, de las cuales 60 se usaron como set de entrenamiento, mientras que las 157 restantes se emplearon para realizar las debidas pruebas, de las cuales se consiguieron los correspondientes resultados. Todas estas imágenes se obtuvieron de forma aleatoria del Atlas Interactivo de la Dermatoscopia, publicado por Edra Medical Publishing New Media [4]. Este atlas interactivo es realmente un proyecto multimedia destinado a la educación médica. Asimismo, dicho Atlas contiene bastantes imágenes de lesiones pigmentadas que provienen de diferentes hospitales y centros médicos, por lo que pueden existir diferentes condiciones de iluminación. Para poder obtener unos resultados fiables, se necesita que un dermatólogo, en este caso un dermatólogo experimentado en el diagnóstico de lesiones pigmentadas, identifique para cada lesión los colores presentes en ella, así como en qué sitio están localizados para poder obtener sus correspondientes máscaras de segmentación.

Por una parte, las 60 imágenes que conforman el conjunto o set de entrenamiento se utilizaron para la extracción manual, por parte de un dermatólogo, de 144 muestras o parches de tamaño 40x40, es decir se sacaron 24 muestras por cada uno de los 6 colores. Dichos parches son usados para la estimación de un modelo de mezcla Gaussiana que se encarga de representar los colores dermatoscópicos (un modelo de mezcla por cada color) [1]. El proceso automatizado para la obtención de estas estimaciones, se ve reflejado en los Códigos 9 y 10 del Anexo A: Códigos . La Figura 4-1 muestra algunos ejemplos de los parches empleados para cada color.

Por otra parte, las 157 imágenes sobrantes componen el conjunto o set encargado de las pruebas. De este conjunto de imágenes, no todas pertenecen a la misma clase: 80 de esas lesiones pertenecen a la clase Melanoma, otras 55 pertenecen a la clase Clark nevus, mientras que las últimas 22 pertenecen a la clase Blue nevus. Cada lesión se segmenta automáticamente gracias al método descrito en el capítulo 3, y el cual se ha implementado en lenguaje Matlab y viene recogido en el Código 12 del Anexo A: Códigos.

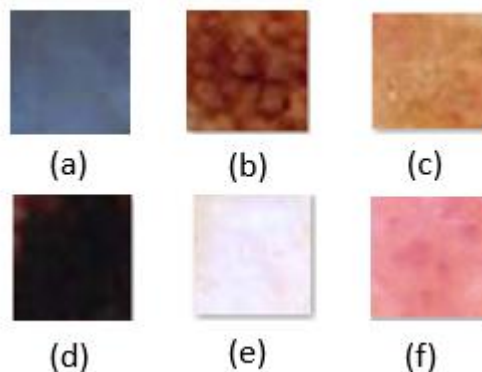


Figura 4-1. Ejemplo de los parches de los 6 colores dermatoscópicos: (a) azul-gris, (b) marrón oscuro, (c) marrón claro, (d) negro, (e) blanco, (f) rojo

4.2 Ajuste y configuración de parámetros del código

En esta sección, se explicará la configuración de los parámetros que se utilizaron para implementar el método propuesto en el capítulo 3, en lenguaje Matlab y así poder automatizar el proceso de detección, identificación y obtención de resultados. Todo el código del que se va hablar en esta sección, se encuentra recogido en el Anexo A: Códigos de este trabajo.

Como son bastantes parámetros y con el fin de que la explicación de sus configuraciones sea lo más clara posible, se dividirá la descripción de dichos ajustes en varias subsecciones correspondientes a determinadas funciones que componen el código presentado. Sólo se describirán las configuraciones de las funciones `Collor_Mixture_Pallete`, `Detect_Colors` y, con el fin de interpretar de una forma más concisa los parámetros de esta última, se hablará también de `Quantify_Colors`. El resto de funciones presentes en el Anexo A: Códigos, se configuran con parámetros ya mencionados en las funciones anteriores, o con otros parámetros que se ajustan dentro del código.

4.2.1 Parámetros de la función `Color_Mixture_Pallete`

Esta función se encarga, en esencia, de extraer los modelos de mezcla gaussiana para cada uno de los colores dermatoscópicos. Para llevar a cabo este propósito, necesita de la correcta configuración de los siguientes parámetros: **X**, **k_max**, **k_min**, **covoption**, **regularize_cov**, **replication**, **color_id**. Esta función está recogida en el Código 1 del Anexo A: Códigos.

El primero de ellos es **X**, que se trata de una estructura de celda que contiene los nombres de los parches de color sacados por el dermatólogo a partir de las 60 imágenes de entrenamiento. Evidentemente existe una **X** para cada color, por lo que la función `Color_Mixture_Pallete` es llamada un total de 6 veces, con el fin de sacar esos 6 modelos de mezcla Gaussiana. Seguidamente, se ajustan los parámetros **k_max** y **k_min**, con los valores 6 y 1 respectivamente. Es importante decir que **k_max** es el número máximo inicial de componentes de la mezcla, mientras que **k_min** es el mínimo número de ellos. A continuación se establece el valor de **covoption**, que controla la matriz de covarianza, y que, como es sabido, puede tomar cuatro valores distintos, desde el 0 hasta el 3 (siendo 0 para covarianzas libres, 1 para covarianzas diagonales, 2 covarianza común para todos los componentes de la matriz y 3 covarianza diagonal común para todos los componentes de la matriz), ambos incluidos. No obstante, como no se han percibido cambios en los resultados empleando un valor u otro de dicho parámetro, **covoption** ha mantenido el valor 0 para todos los casos. Por otra parte, el valor del factor de regularización de estas matrices de covarianza, que viene dado por **regularize_cov**, comparte el mismo motivo que **covoption**, por lo que también su valor es 0. Por último, para el ajuste de **replication** y **color_id**, que son el número de veces que se ejecuta el algoritmo y la etiqueta que identifica al color estimado respectivamente, se usa, en el caso de **replication**, el valor 3 simplemente por defecto (estaría bien incluso si no se usara este parámetro ya que la mejor muestra es dada en la primera iteración). Sin embargo, con respecto a **color_id**, se utilizan las siguientes etiquetas para cada color: BG para el azul-gris, DB para el marrón oscuro, LB para el marrón claro, B para el negro, W para el blanco y R para el rojo.

4.2.2 Parámetros de las funciones `Detect_Colors` y `Quantify_Colors`

En esta subsección se hará una descripción de las funciones `Detect_Colors` y `Quantify_Colors` ya que ambas funciones están directamente relacionadas debido a que dentro de la primera de ellas se llama a la segunda. Estas funciones están recogidas en los Códigos 2 y 3, respectivamente, del Anexo A: Códigos.

En primer lugar, la función `Detect_Colors` se encarga de determinar las máscaras (**BG**, **DB**, **LB**, **B**, **W**, **R**) para cada color que compone la imagen evaluada, además de proporcionar la máscara global de etiquetado de los parches de color (**M**). Asimismo, los parámetros que `Detect_Colors` utiliza para conseguir los valores anteriores (**BG**, **DB**, **LB**, **B**, **W**, **R** y **M**) son los siguientes: **img**, **step**, **seg**. La configuración de estos parámetros es muy sencilla. Primeramente, el parámetro **img** es la imagen que va a ser analizada y que está en formato uint8. Seguidamente, el parámetro **step**, que viene configurado con el valor 1 por defecto, siendo este su valor óptimo para el etiquetado de un pixel a la vez, es el tamaño de los parches que van a ser clasificados. Finalmente **seg**, es la máscara de segmentación binaria de la lesión. Dicha máscara, es proporcionada junto a las imágenes que

componen la base de datos. Como se observa en el Código 12 del Anexo A: Códigos, la máscara de segmentación proporcionada no es en primera instancia binaria, por lo que para poder emplear la función `Detect_Colors` se convierte a binaria.

Por otra parte, la función `Quantify_Colors` es la responsable de obtener los colores que han sido detectados en cada una de las imágenes que componen la base de datos (**detected_colors**), así como de la adquisición de las máscaras de color (**color_mask**) de los pigmentos que conforman las imágenes analizadas, a través de la cuantificación de los mismos usando los modelos de mezcla Gaussiana descritos en el capítulo 3. Los parámetros que utiliza `Quantify_Colors` para obtener los valores anteriores (**detected_colors** y **color_mask**), son las salidas de la función `Detect_Colors`, es decir, **BG, DB, LB, B, W, R** y **M**.

Antes de concluir esta subsección, hay que resaltar que en la función `Detect_Colors`, con objeto de obtener las máscaras de cada color, se hace una corrección del espacio RGB y posteriormente se emplean dos espacios de color, HSV y La^*b^* por separado y de forma conjunta alternativamente.

4.3 Resultados de la identificación de colores en las lesiones

En esta sección se realizará una explicación exhaustiva del procedimiento empleado para hallar los resultados obtenidos del análisis y evaluación de las imágenes, así como la descripción del significado de los mismos. Dichos resultados quedarán recogidos en dos tablas muy diferentes, que se desarrollarán en las subsecciones posteriores, y que servirán para evaluar el algoritmo según diferentes puntos de vista, con objeto de comprobar su eficacia. Además, en esta sección se comentará como se han extraído las máscaras de segmentación para cada color correspondientes a las imágenes proporcionadas por el dermatólogo. Antes de nada hay que especificar que ambas tablas expuestas muestran sus resultados con cuatro decimales de precisión para intentar conseguir unos resultados más exactos.

4.3.1 Obtención de las máscaras de segmentación de las imágenes evaluadas

El paso de obtener las máscaras de segmentación de cada color que conformara la imagen proporcionada y evaluada debidamente por el dermatólogo, es un paso necesario ya que, sin dichas máscaras, la obtención de los TP (True Positives), TN (True Negatives), FP (False Positives), FN (False Negatives), (a través de los cuáles se calculan los parámetros recogidos en las tablas presentadas en las subsecciones inferiores) no sería posible. En este trabajo, la extracción de dichas máscaras se ha hecho a través de la aplicación `Color Thresholder`. Esta aplicación, permite ajustar las imágenes en color mediante la manipulación de los componentes de color existentes en las mismas, en función de los diversos espacios de color, permitiendo así poder utilizarla para crear máscaras de segmentación de este tipo de imágenes. A continuación, se hará una descripción, en varios pasos, del proceso de obtención de las máscaras de segmentación necesarias para analizar la viabilidad del algoritmo.

4.3.1.1 Pasos a seguir para la obtención de máscaras de segmentación binarias

Para poder extraer las máscaras de segmentación de los colores presentes en imágenes dermatoscópicas, en concreto, con la aplicación `Color Thresholder`, se deben seguir los siguientes pasos, que son los mismos que se han utilizado en este trabajo y que se describen a continuación.

1. Ejecución de la aplicación en Matlab y presentación de su interfaz

Se comienza el análisis con la ejecución de la aplicación en Matlab mediante el comando `colorThresholder`. Al introducir dicho comando y darle a enter, la aplicación se abre al instante mostrando la interfaz. Esta interfaz está compuesta por varias secciones (siete en total), donde los componentes de las mismas vienen descritos en la línea siguiente, y numerados de izquierda a derecha en la Figura 4-2. En la primera de ellas (1), encargada de la obtención de imágenes, encontramos una herramienta para importar las imágenes al espacio de trabajo, llamada `Load image`. En la sección consecutiva (2), que se hace cargo de los espacios de color, se encuentra, de igual manera que en la anterior, una herramienta llamada `New Color Space`, que, en este caso, permite seleccionar el espacio de color deseado de entre cuatro posibles: RGB, HSV, YCbCr, La^*b^* . En tercer lugar (3), se encuentra la sección de selección de color, la cual contiene dos herramientas, la primera de ellas para poder

seleccionar el color o colores deseados (Select Colors) y la segunda para obtener esas regiones que presentan los colores seleccionados previamente por la herramienta anterior (Find Thresholder). Seguidamente, se pueden observar tanto la sección que se encarga de la inversión de la máscara (4) (cambiar de ceros a unos y viceversa) y que tan solo presenta una herramienta (Invert Mask), y la sección que contiene herramientas de zoom y movimiento (5) (Zoom in, Zoom out, Pan). La sexta sección (6) que presenta esta interfaz, es la que se encarga de la segmentación y contiene tres herramientas, dos de ellas relacionadas con propiedades del fondo de la imagen (Background Color y Background Opacity) y una última que proporciona la máscara de segmentación en formato binario. Finalmente, la última de las secciones (7) se encarga de la exportación de la máscara obtenida a través de un único botón (Export)



Figura 4-2. Interfaz ampliada de la aplicación Color Thresholder en la que se numeran todas las secciones de las que está compuesta.

2. Importación de imágenes y selección del espacio de color

El siguiente paso a seguir, después de la ejecución de la aplicación, es la selección de la imagen que se va a segmentar, así como del espacio de color en el que se va a trabajar. En este documento, el espacio de color elegido, a la hora de segmentar las imágenes utilizando esta aplicación, es RGB. En la Figura 4-3 se puede observar la importación de imágenes, mientras que en la Figura 4-4 se puede apreciar el proceso de selección del espacio de color que surge de forma automática al cargar la imagen, aunque también se puede seleccionar de forma manual mediante la herramienta New Color Space mencionada en el punto anterior. Finalmente, una vez realizado el proceso de importación y elegido el espacio de color (RGB), la imagen está lista para ser segmentada en los colores requeridos. Este entorno de trabajo listo para la segmentación se presenta en la Figura 4-5.

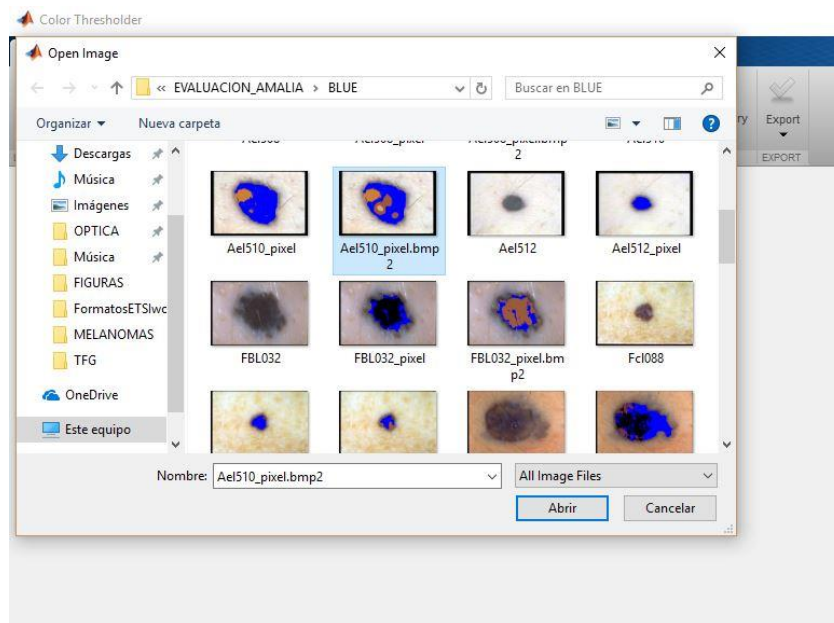


Figura 4-3. Importación de la imagen Ael510_pixel.bmp2 de la clase Blue nevus que contiene los colores detectados por el dermatólogo que forman las máscaras de segmentación

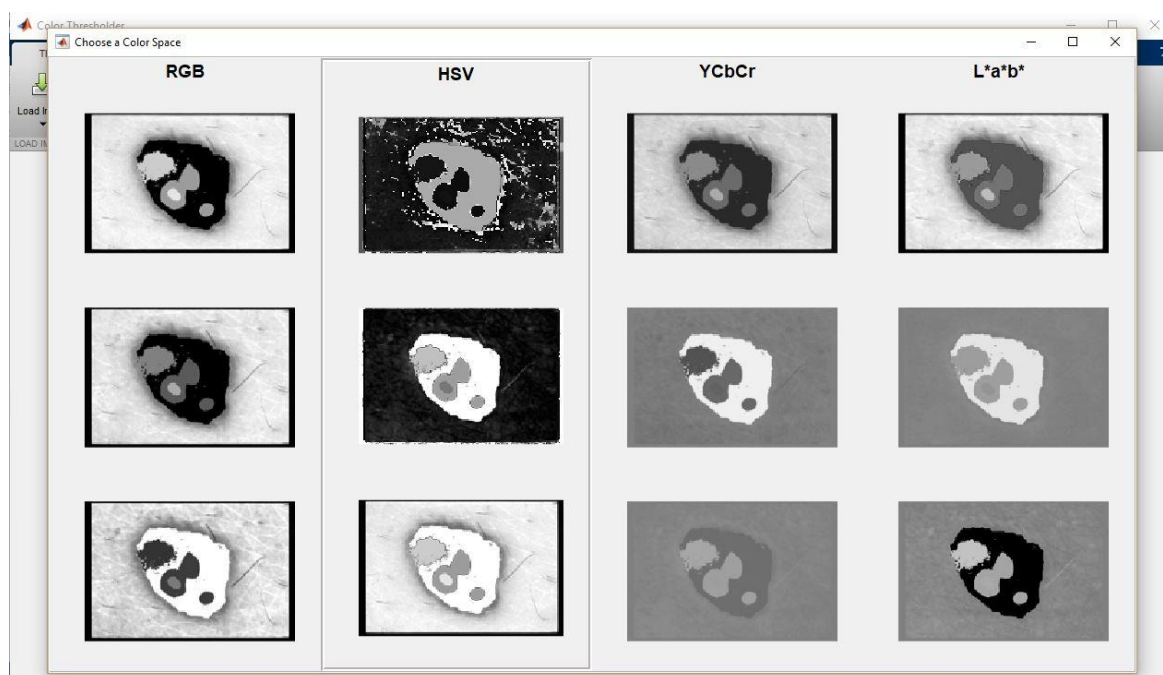


Figura 4-4. Selección del espacio de color RGB.

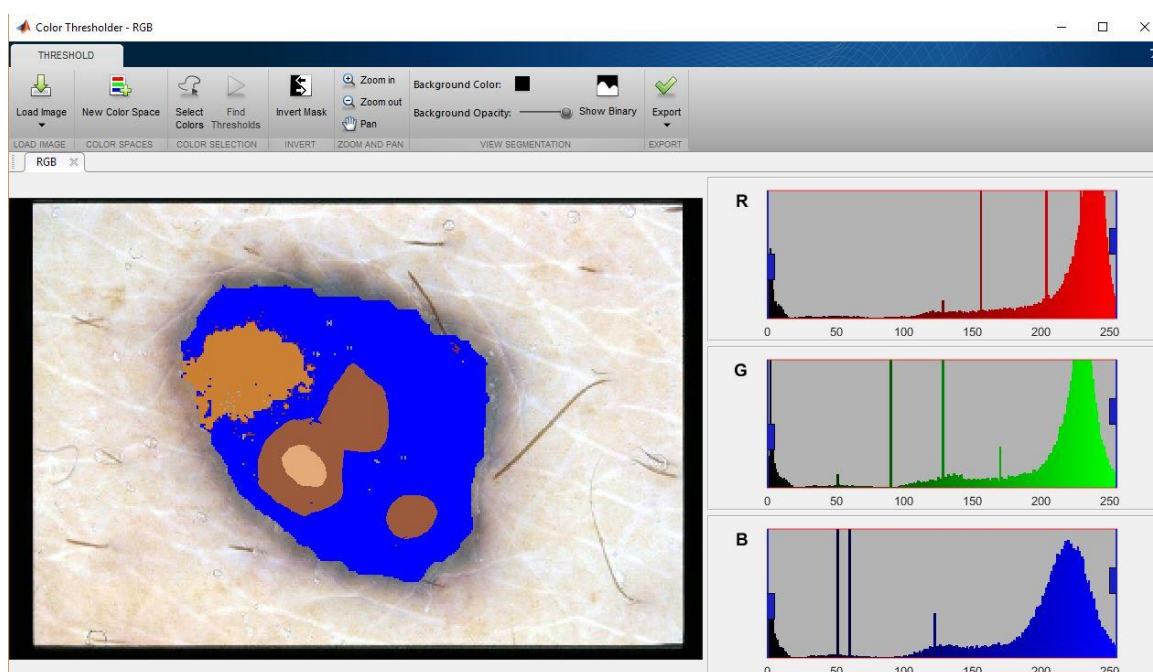


Figura 4-5. Entorno de trabajo final, listo para la segmentación de la imagen en los colores detectados.

3. Segmentación de imágenes y obtención de máscaras

Una vez elegido el espacio de color, con el que se va a trabajar, se segmentan los pigmentos presentes en la imagen, los cuales han sido clasificados por el especialista, consiguiendo así las máscaras binarias de segmentación para cada uno de ellos. Estas máscaras se compararán en las subsecciones posteriores con las de los colores detectados por el algoritmo propuesto, cuya metodología ha sido explicada en el capítulo 3. En la Figura 4-6 se muestra el proceso de selección del color a segmentar, en este caso el azul, a través de Select Colors (mediante un círculo blancuino que selecciona el área de color deseada), y posteriormente, en la Figura 4-7, se muestra la obtención de dicho color por separado. Finalmente, en la Figura 4-8, se presenta la máscara binaria del color segmentado, extraída gracias a la tercera herramienta de la penúltima sección de la interfaz expuesta.

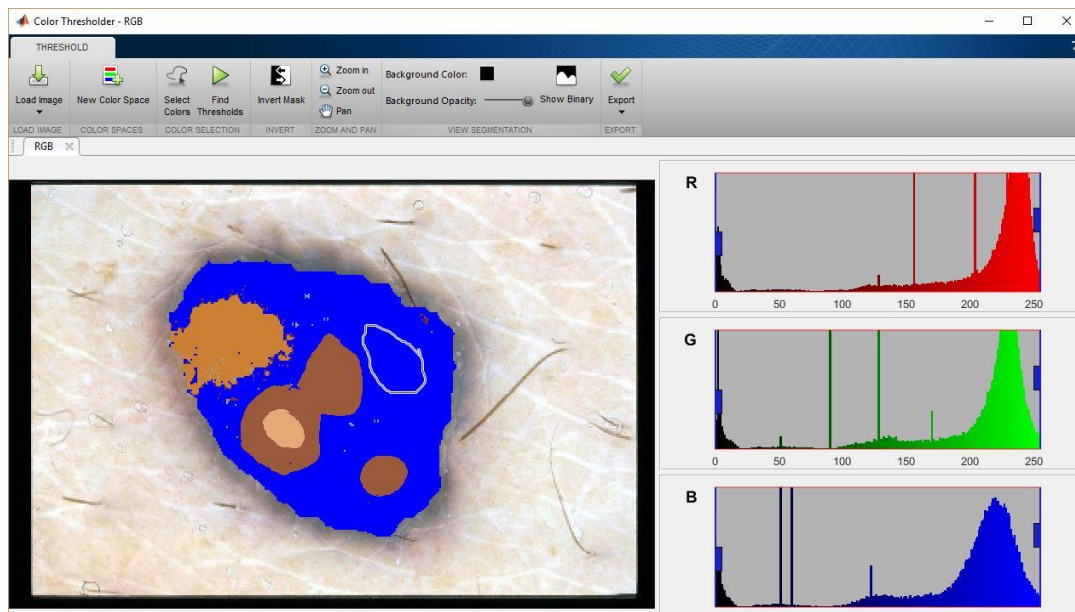


Figura 4-6. Proceso de selección del color azul, a través de la herramienta Select Colors, para su posterior segmentación.

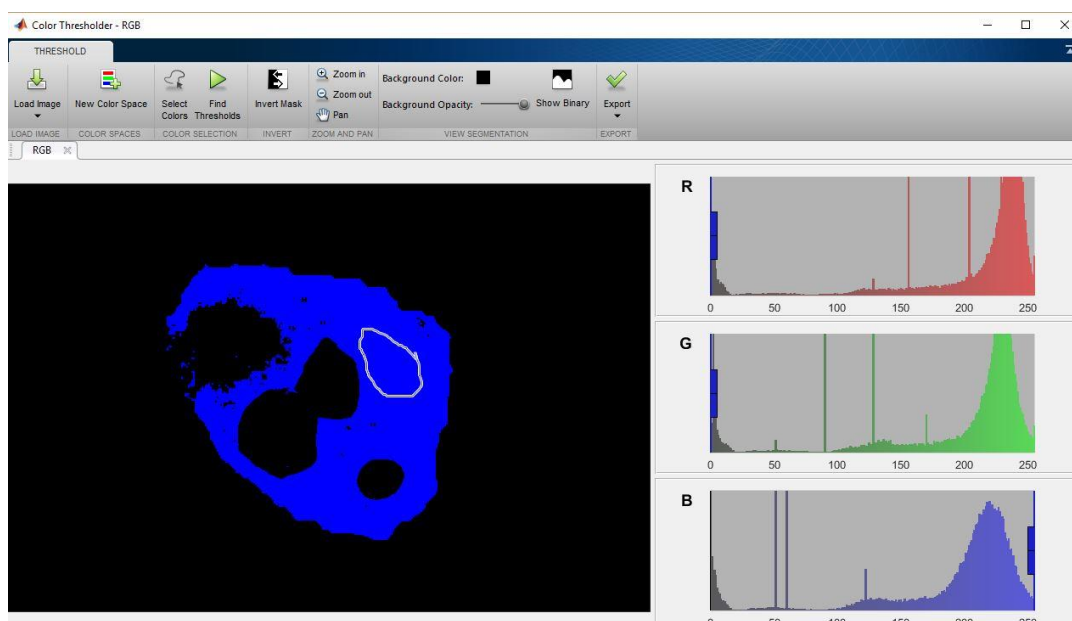


Figura 4-7. Proceso de separación del color azul del resto de colores a través de la herramienta Find Thresholds.

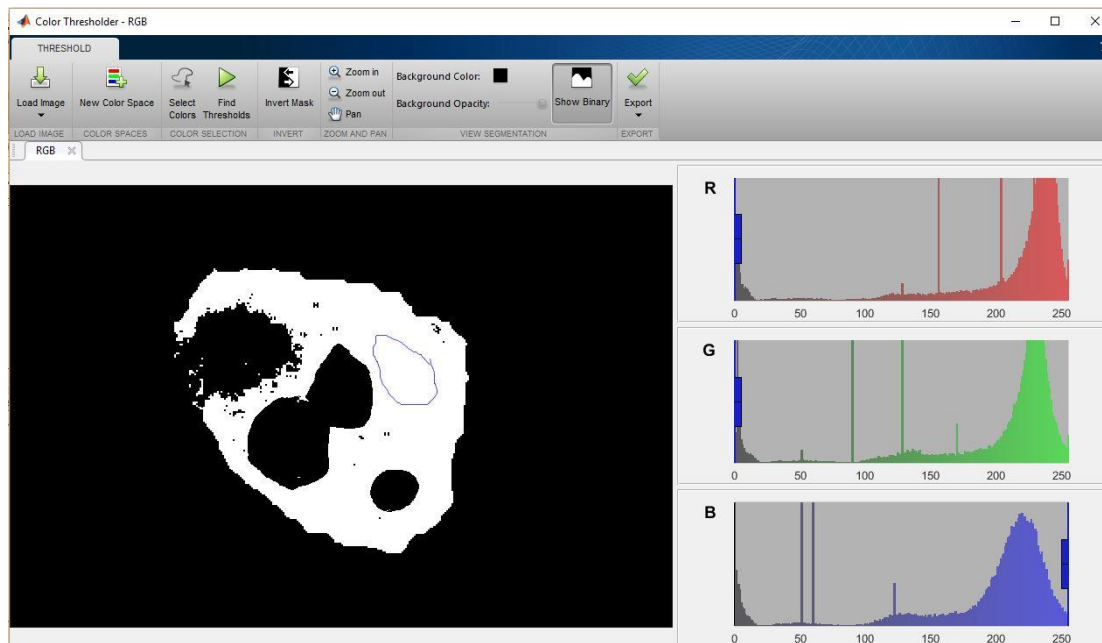


Figura 4-8. Proceso de obtención de la máscara de segmentación binaria a través de la herramienta Show Binary.

4. Exportación de imágenes al espacio de trabajo de Matlab

Ya obtenida la máscara binaria del color deseado, esta se exporta al entorno de trabajo de Matlab, tal y como se muestra en la Figura 4-9, en la que como se puede observar, es posible cambiar el nombre a la máscara del color azul para así hacer más fácil su uso posterior. En este caso, al ser el color azul el que ha sido segmentado, para guardar su máscara, lo único que se hace es añadir “_azul” al nuevo nombre de la máscara.

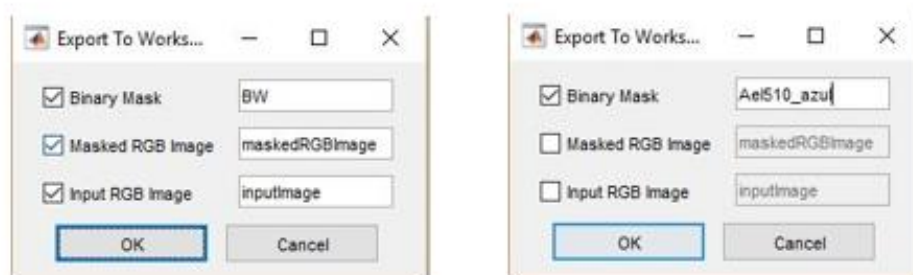


Figura 4-9. Exportación de la máscara binaria del color azul al entorno de trabajo de Matlab.

5. Guardar imágenes en el dispositivo

Finalmente, para registrar la máscara binaria y poder emplearla en futuras comparaciones (obteniendo así los resultados requeridos), esta ha de ser guardada en alguna carpeta del dispositivo. Dicho proceso se muestra de forma gráfica en la Figura 4-10. Además en la Figura 4-11 se puede observar cómo sería la imagen analizada en estos puntos si se segmentara en las máscaras de todos los colores que la componen.

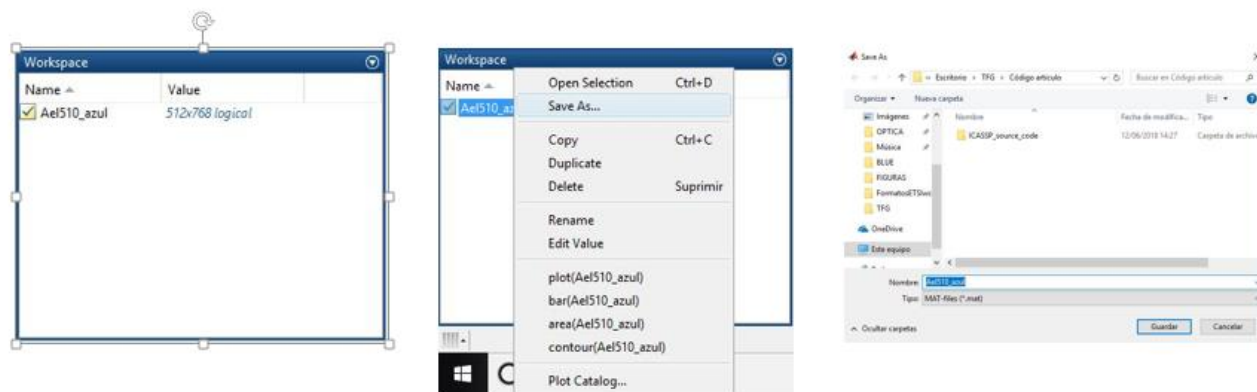


Figura 4-10. Proceso de guardado de imágenes a partir del entorno de trabajo de Matlab.

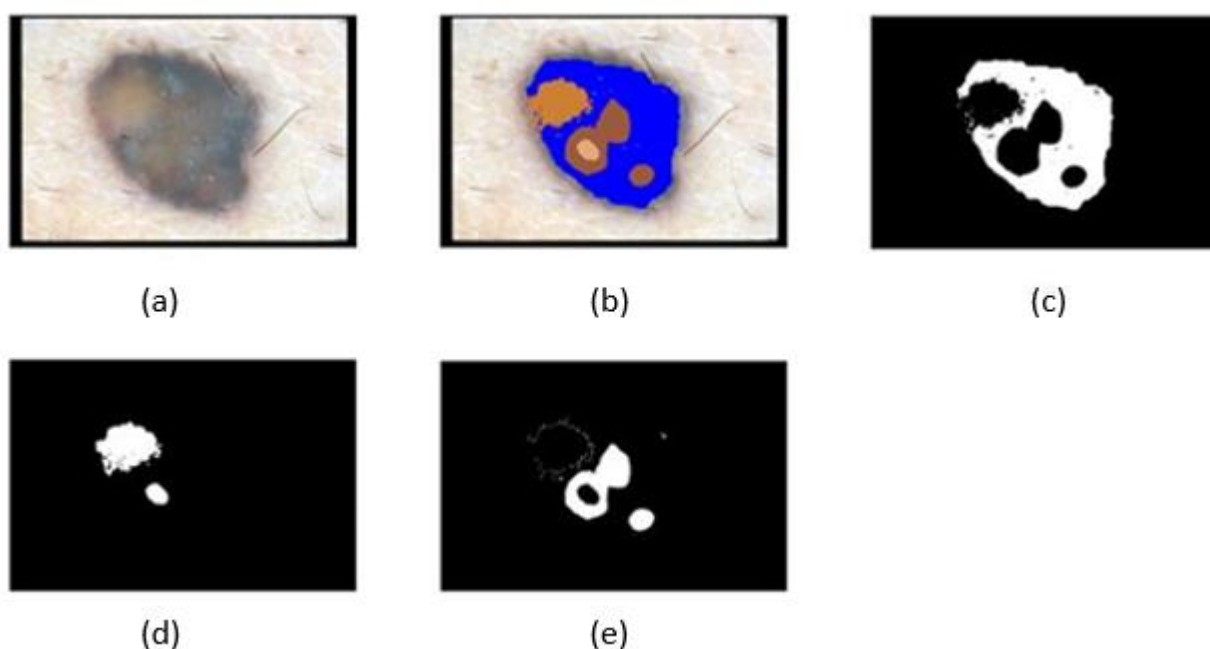


Figura 4-11. Proceso de segmentación completa de la imagen Ael510 de la clase Blue nevus: (a) Imagen original, (b) Imagen con las máscaras de color detectadas por el dermatólogo, (c) Imagen de la segmentación del color azul presente en la imagen (b), (d) Imagen de la segmentación del color marrón claro presente en la imagen (b), (e) Imagen de la segmentación del color marrón oscuro presente en la imagen (b).

4.3.2 Evaluación de la asignación de colores a los píxeles de las lesiones

En esta subsección se hará un análisis, por el algoritmo propuesto, de los colores asignados a los píxeles de cada una de las lesiones que componen la base de datos de imagen. De esta manera, se evaluará si dichos píxeles han sido clasificados correctamente, mediante la comparación de las máscaras de los colores asignados a los mismos, con las máscaras detectadas por el dermatólogo.

4.3.2.1 Procedimiento de comparación de máscaras y obtención de TP, TN, FP y FN

Para poder obtener los resultados que se verán en la Tabla 4-1, se han de comparar las máscaras de segmentación de los colores asignados a los píxeles que conforman la lesión analizada, detectados mediante el algoritmo propuesto y las máscaras de segmentación de los aquellos clasificados por el dermatólogo. Para poder realizar dicha comparación, se ha fabricado un código en Matlab que queda reflejado en los Códigos 11 y 12 del Anexo

A: Códigos, donde se puede observar cómo se han obtenido los parámetros TP, TN, FP y FN, conseguidos al cotejar ambas segmentaciones (la del algoritmo y la del especialista). Se puede observar en la Figura 4-12 un pequeño dibujo en el que se muestra la clasificación de las variables anteriores. Dichos parámetros sirven para poder validar un algoritmo y tienen el siguiente significado:

- TP: representa los píxeles de la intersección entre las dos segmentaciones, es decir tanto el algoritmo como el especialista coinciden.
- TN: representa los píxeles que no han sido segmentados ni por el algoritmo ni por el especialista.
- FP: representa los píxeles segmentados por el algoritmo y no por el especialista.
- FN: representa los píxeles que pertenecen a la segmentación del especialista pero que no han sido segmentados por el algoritmo.

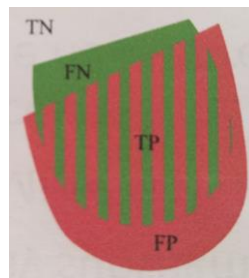


Figura 4-12. Esquema de variables TP, TN, FP, FN

Una vez conseguidos los valores de estas variables al comparar ambas máscaras de segmentación (hay que aclarar que se obtendrán esos cuatro valores para cada comparación realizada de las máscaras, siempre comparándose las del mismo color a la vez), se han de calcular ciertos parámetros especiales que se encargarán de evaluar la calidad de la segmentación realizada. Pero para poder obtener dichos parámetros, es necesario que se calculen los cuatro valores anteriores para todas las imágenes que formen la base de datos. Cuando esas estimaciones, en sus versiones totales, sean extraídas para cada uno de los colores dermatoscópicos, se pasa al cálculo de los parámetros que se describirán en el siguiente apartado. Es importante resaltar que los valores de TP, TN, FP y FN en sus formatos tanto totales como individuales (para cada imagen), se pueden encontrar en la Tabla 2 del Anexo B: Datos Relevantes.

4.3.2.2 Obtención de parámetros que miden la calidad de la segmentación

Los parámetros que van a servir para poder medir y evaluar la calidad de la segmentación, y que, como se ha mencionado en el apartado anterior, han sido obtenidos mediante los valores totales de TP, TN, FP y FN, son los siguientes:

- Sensibilidad (Sen): capacidad que mide qué porcentaje de la segmentación real ha segmentado el algoritmo. Este parámetro alcanzará su mayor valor, es decir 1 o 100 (si se mide en porcentaje), cuando no se detecten FN, lo que implica que todos los píxeles han sido correctamente clasificados, en otras palabras todas las anomalías (eventos) son detectadas. En cambio, dicho parámetro tomará valores próximos a 0, su valor mínimo, si existen zonas con muchos FN, lo que lleva a que los píxeles que conforman dichas regiones, no han sido correctamente clasificados [27], [28]. Si es baja hay subsegmentación.

$$Sen = \frac{TP}{TP + FN} \quad (4-1)$$

- Valor predictivo positivo (VPP): cifra mide qué porcentaje de los píxeles segmentados por el algoritmo pertenecen realmente a la región segmentada por el especialista, es decir se encarga de evaluar la eficiencia real del algoritmo [27]. Tomará valores cercanos a 1 o 100 (en términos de porcentaje), cuando no haya presencia de FP, mientras que alcanzará valores próximos a 0 cuando haya regiones

con más FP que TP. Si es baja significa que habrá sobresegmentación.

$$VPP = \frac{TP}{TP + FP} \quad (4-2)$$

- Exactitud (Acc): parámetro que, según [29], indica en términos de porcentaje la precisión de la segmentación realizada, es decir, mide la proporción de píxeles etiquetados correctamente como lesión por el algoritmo, entre el total de píxeles presentes en la imagen [27].

$$Acc = \frac{TP + TN}{TN + TP + FP + FN} \quad (4-3)$$

4.3.2.3 Análisis de resultados de la Tabla 4-1

A partir de los parámetros que se explican en el apartado de anterior, se presentan los siguientes resultados para cada uno de los seis colores dermatoscópicos. Estos resultados serán analizados para poder evaluar la eficiencia del algoritmo, así como la calidad de la segmentación que el mismo ofrece. Los valores obtenidos se muestran en la Tabla 4-1.

Tabla 4-1. VPP, Sen, Acc de la asignación de colores a los píxeles de las lesiones pigmentadas, en tanto por uno

Colores	VPP	Sen	Acc
Negro	0,1233	0,9772	0,8814
Azul-Gris	0,5124	0,5283	0,9318
Marrón Oscuro	0,3049	0,2788	0,8838
Marrón Claro	0,5022	0,4880	0,9119
Rojo	0,4027	0,8054	0,9766
Blanco	0	0	0
Total	0,2984	0,4406	0,9003

Para poder discurrir mejor el significado de cada uno de los valores presentados en la tabla superior, se analizarán y explicarán las interpretaciones de los parámetros de calidad de los colores por separado. Por último, se expondrá el alcance del algoritmo al poder extraer conclusiones de los valores finales de dichos parámetros de calidad. Sin embargo, cabe destacar que no todos los colores han sido detectados en la misma medida, es decir, colores como el negro, azul-gris, marrón oscuro, marrón claro fueron identificados en un gran número de lesiones, a diferencia de los colores rojo y blanco que fueron detectados en muy pocas lesiones. Esto se debe, en esencia, a que la detección de estos dos últimos colores es una tarea bastante complicada, dado que las muestras obtenidas para el enteramiento del algoritmo no siempre son de la mejor calidad, ya que la presencia del color rojo está condicionada por la aparición de capilares, mientras que la aparición del color blanco solo es considerada si hay partes de la lesión mucho más claras que la piel adyacente.

Antes de comenzar con la explicación de los valores obtenidos, es importante saber que el algoritmo presentado en este trabajo consigue máscaras de segmentación que abarcan un área mayor que las sacadas por el especialista. Este factor, por tanto, puede suponer una posible fuente de error que lleve a que los resultados expuestos en la Tabla 4-1 no sean del todo exactos. En la Figura 4-13 se tiene un ejemplo de esta situación.



Figura 4-13. Ejemplo de la segmentación de la lesión Ael446 de la clase Blue nevus perteneciente a la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color azul de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color azul realizada por el algoritmo. Se observa como (d) abarca mayor área que (c).

A continuación se realizará el análisis mencionado en este párrafo:

- **Negro**

Como se puede observar en la Tabla 1 del Anexo B: Datos Relevantes, el color negro está entre los más detectados en las lesiones de la base de datos. Este color, es detectado por el algoritmo propuesto con demasiada asiduidad y, en varias ocasiones, es identificado incluso cuando el dermatólogo no lo detecta, por lo que puede inducir a varios errores. Esto queda reflejado en la Figura 4-14 y en la Figura 4-15. En esta última se comprueba que aunque el especialista detecta solamente el color azul, el algoritmo identifica además el color negro. También es importante destacar que las máscaras de segmentación del negro, realizadas por el algoritmo propuesto, al ser comparadas con las extraídas por el especialista, presentan un área mayor que las del médico, e incluso cuando los colores detectados por éste y el algoritmo propuesto coinciden, una gran parte de las veces, es el color negro el que posee una máscara de segmentación mucho mayor que ninguna de otro color presente en la lesión. Este fenómeno se muestra en la Figura 4-16 y la Figura 4-17.

	EVALUACIÓN ALGORITMO							EVALUACIÓN DEL ESPECIALISTA					
	Negro	Azul	Dark	Light	Red	White		Negro	Azul	Dark	Light	Red	White
Gdl037'	1	1	0	0	0	0		0	0	1	1	1	0
'Ggl006'	1	1	0	0	0	0		0	1	0	0	0	0
'Nml089'	1	1	0	0	0	0		0	1	0	0	0	0
'Aal012'	1	1	1	0	0	0		0	1	1	1	0	0

Figura 4-14. Comparación entre la evaluación del algoritmo y la evaluación del especialista en lo referente a la presencia del color negro en las lesiones analizadas (Gdl037 perteneciente a la clase Melanomas, Ggl006 y Nml089 pertenecientes a la clase Blue nevus y Aal012 perteneciente a la clase Clark nevus). Todas las lesiones presentadas en la imagen muestran la alta presencia del negro por encima de la detección de otro color (Gdl037, Aal012), no coincidiendo así con la detección realizada por el especialista.

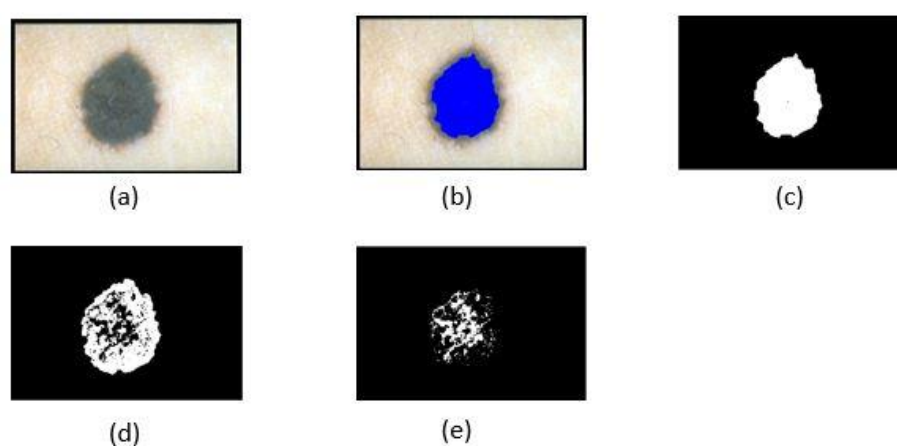


Figura 4-15. Comparación y detección del color negro en la lesión Nml089 de la clase Blue nevus de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el

dermatólogo, (c) Máscara de segmentación del color azul de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color azul realizada por el algoritmo, (e) Máscara de segmentación del color negro realizada por el algoritmo y no detectado por el especialista

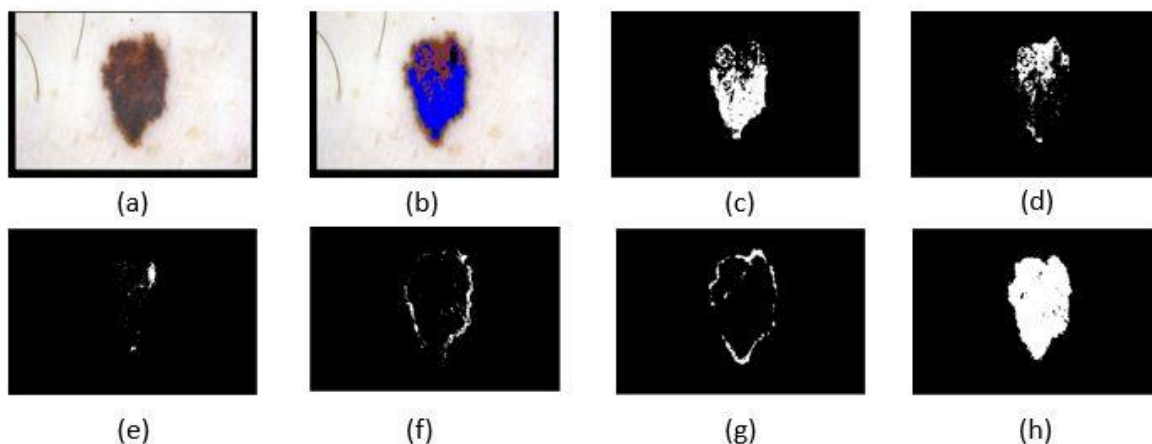


Figura 4-16. Comparación y detección del color negro en la lesión Aal010 de la clase Clark nevus de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color azul de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color marrón oscuro de (b) realizada por el dermatólogo, (e) Máscara de segmentación del color negro de (b) realizada por el dermatólogo, (f) Máscara de segmentación del color azul realizada por el algoritmo, (g) Máscara de segmentación del color marrón oscuro realizada por el algoritmo, (h) Máscara de segmentación del color negro realizada por el algoritmo, presentando una mayor área que (e).

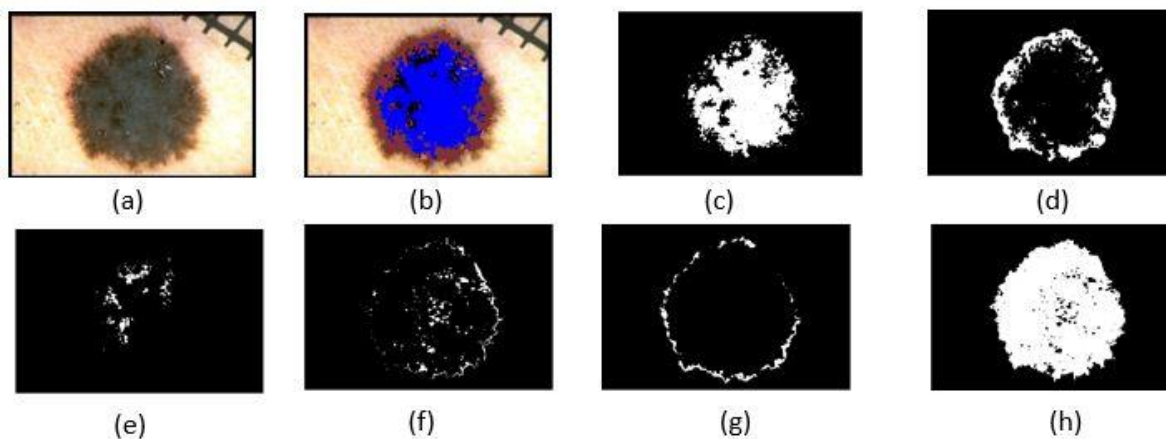


Figura 4-17. Comparación y detección del color negro en la lesión Ngl021 de la clase Melanoma de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color azul de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color marrón oscuro de (b) realizada por el dermatólogo, (e) Máscara de segmentación del color negro de (b) realizada por el dermatólogo, (f) Máscara de segmentación del color azul realizada por el algoritmo, (g) Máscara de segmentación del color marrón oscuro realizada por el algoritmo, (h) Máscara de segmentación del color negro realizada por el algoritmo, presentando una mayor área que (e).

En cuanto a los valores obtenidos para los parámetros que evalúan la calidad de la segmentación del color negro, se pueden sacar varias conclusiones. En primer lugar, con respecto al VPP, se puede decir que presenta un valor muy pequeño (el más bajo obtenido para este parámetro en la Tabla 4-1), tal y como se puede ver en la Tabla 4-1, lo que indica que hay sobresegmentación para dicho color. Como se explica en [30], la sobresegmentación de una región, se da cuando varias regiones de la imagen segmentada de forma automática corresponden a una sola región de la imagen de referencia. Es decir, varias regiones de la imagen segmentada por el algoritmo se corresponden con tan solo una región de la imagen segmentada por el dermatólogo. Por tanto, lo que se comentaba en el párrafo anterior es realmente la causa de este efecto. Al albergar el color negro una mayor área en la segmentación hecha por el algoritmo, causa, que al compararse con la muestra realizada por el especialista, provoque una clara sobresegmentación.

Por otra parte, al examinar el parámetro de Sensibilidad (Sen) se puede observar que es bastante alto acercándose casi a la unidad, luego se puede descartar totalmente la subsegmentación, y concluir que para el color negro, el algoritmo, en términos de porcentaje, segmenta un 97,72 % de la segmentación real (clasificación correcta de la segmentación real). Es decir, como se establece en [28], una sensibilidad que se acerca al 100% indica que se detectan casi todas las anomalías que se puedan dar en la imagen, si es que existen realmente.

Siguiendo con el análisis, finalmente se expone el último de los parámetros evaluados, la llamada Exactitud (Acc), que como se puede notar en los resultados expuestos en la Tabla 4-1, su valor es muy elevado, y esto indica que la segmentación obtenida para el color negro es bastante buena. Se conseguiría una segmentación perfecta si el valor de dicho parámetro fuera la unidad, como se mencionaba anteriormente.

- **Azul-Gris**

Al igual que el color negro, el azul-gris es un color que ha sido detectado en casi todas las imágenes que componen las distintas clases dadas por la base de datos. Sobre todo, hay que destacar que en la clase Blue nevus, en todas las imágenes excepto en una, FBL032, el color azul es identificado por el algoritmo propuesto. Esto último puede ser observado en las Tablas 1 y 2 del Anexo B: Datos Relevantes.

Sin embargo, la identificación de la presencia de este color en un principio no se pudo realizar de manera exitosa, ya que se poseían unos datos de entrenamiento que no conseguían que el algoritmo detectara este color en las lesiones segmentadas. Por tanto, se cambiaron 24 muestras de parches para el azul-gris por otras 24 que se acercaban mucho más a la representación de un modelo capaz de detectar dicho color en las lesiones propuestas. Esto deja entre ver la variabilidad de los resultados obtenidos por el algoritmo. En la Figura 4-18 se pueden observar algunos de los parches sustituidos, los cuales presentaban un subtono más oscuro haciendo que las lesiones que presentaban áreas azul-grises, fueran clasificadas como negras en su totalidad.

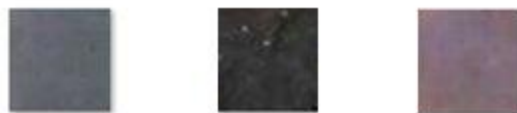


Figura 4-18. Parches antiguos del color azul-gris

Por otra parte, con respecto a los valores obtenidos para los parámetros que evalúan la calidad de la segmentación del color azul-gris, se pueden sacar varias conclusiones. En primer lugar, en cuanto al VPP, a diferencia del color negro, un poco más de la mitad de los píxeles son FP. Sin embargo, aunque el valor de este parámetro no sea tan bajo como en el caso del negro, no se puede descartar la sobresegmentación. En la Figura 4-19 se muestra una prueba de ello.

Seguidamente, se puede observar que al examinar el parámetro de Sensibilidad (Sen), su valor respecto al negro descende situándose en casi la mitad, tal y como se puede comprobar en la Tabla 4-1. No obstante, sigue siendo un valor bajo en términos de Sensibilidad por lo que existe subsegmentación. Este fenómeno se puede observar en la Figura 4-20.

El último de los parámetros en ser analizados es la Exactitud (Acc), que en este caso es mucho más alta conseguida por el color negro. Este valor tan alto, se puede interpretar como que la segmentación que realiza el algoritmo para el color azul-gris es muy precisa (el porcentaje de precisión de la segmentación, como se observa en la Tabla 4-1, es del 93,18%)

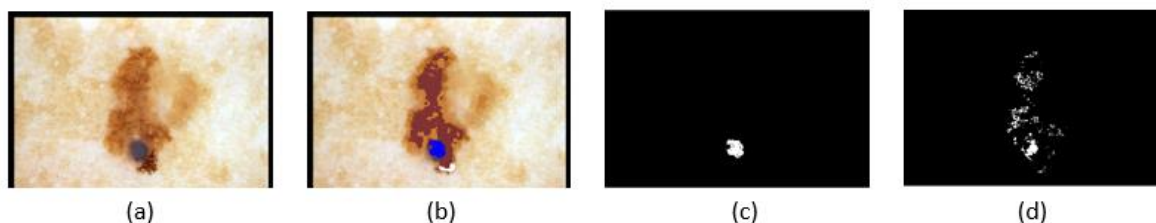


Figura 4-19. Sobresegmentación del color azul-gris en la lesión Ael490 perteneciente a clase Melanoma de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color azul-gris de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color azul-gris realizada por el algoritmo.

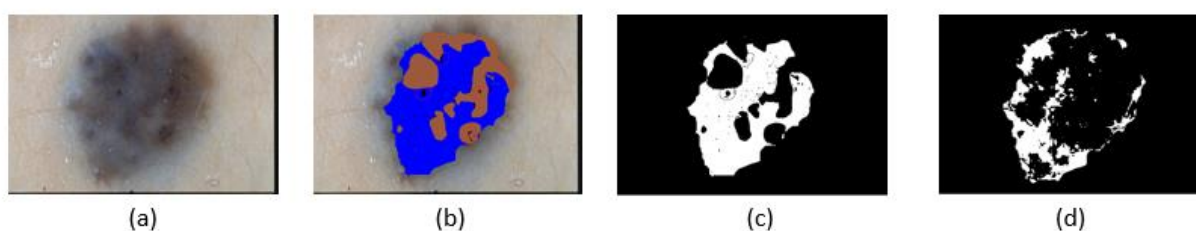


Figura 4-20. Subsegmentación del color azul-gris en la lesión Fgl063 perteneciente a clase Blue nevus de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color azul-gris de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color azul-gris realizada por el algoritmo.

- **Marrón Oscuro**

Este color, al igual que los dos anteriores, está entre los pigmentos más detectados dentro de las lesiones presentadas. Sobre todo, se puede apreciar una fuerte presencia del marrón oscuro en la clase Clark nevus de la base de datos, y una menor presencia en el resto de clases, tal y como se puede comprobar en las Tablas 1 y 2 del Anexo B: Datos Relevantes.

Pasando al análisis de los parámetros, se puede decir, en primer lugar, que el valor obtenido para VPP es más bajo que el conseguido en el caso del color azul-gris, pero más alto que el extraído para el negro. Por tanto, la sobresegmentación no puede ser descartada en este caso, ya que la cifra presentada no es especialmente alta. En la Figura 4-21 se muestra una prueba de esto.

Con respecto al valor conseguido para el parámetro de Sensibilidad (Sen), es importante destacar que el número conseguido para este caso es el más bajo entre los colores evaluados (descartando por supuesto el blanco, que no se pudo analizar), y al situarse en torno al 0,2788, como se muestra en la Tabla 4-1, indica claramente que existe subsegmentación. La sub-segmentación de una región se da,

según [28], cuando varias regiones de la imagen segmentada por el especialista corresponden a una sola región de la imagen segmentada por el algoritmo de forma automática. Este fenómeno se muestra en la Figura 4-22.

Para terminar el análisis del marrón oscuro, se interpreta el valor obtenido por el parámetro Exactitud (Acc), que al presentar un valor bastante alto, aunque un poco más bajo que en el caso del negro, indica que la precisión que presenta el algoritmo al realizar la segmentación es bastante buena.



Figura 4-21. Sobresegmentación del color marrón oscuro en la lesión Fhl066 perteneciente a clase Clark nevus de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color marrón oscuro de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color marrón oscuro realizada por el algoritmo.

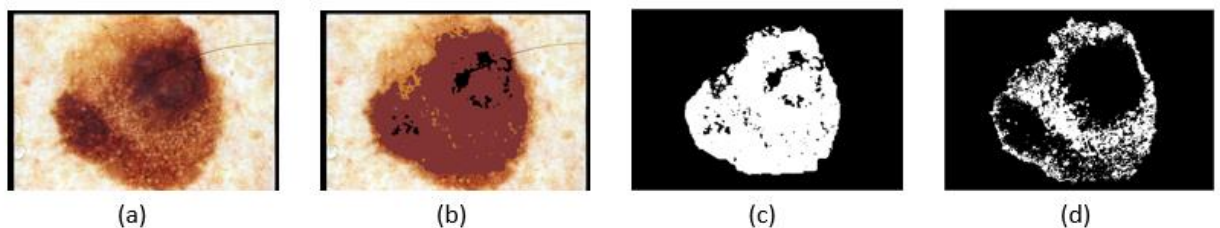


Figura 4-22. Subsegmentación del color marrón oscuro en la lesión Acl039 perteneciente a clase Clark nevus de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color marrón oscuro de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color marrón oscuro realizada por el algoritmo.

• Marrón Claro

A diferencia de los tres colores anteriores, el color marrón claro se detecta con menor frecuencia por el algoritmo propuesto. Además, tal y como se puede observar en las Tablas 1 y 2 del Anexo B: Datos Relevantes, en la clase Blue nevus, no se detecta ni una sola vez, siendo, por el contrario, en la clase Clark nevus donde es identificado con mayor asiduidad, sin ser tampoco un gran número de veces.

En cuanto a los valores obtenidos para los parámetros que se encargan de medir la calidad de segmentación del color marrón claro, se pueden extraer las siguientes conclusiones. Primeramente, al examinar el valor del parámetro VPP, se puede notar que el algoritmo es capaz de segmentar un 50,22% de los píxeles que pertenecen realmente a la región segmentada por el especialista. Ese valor indica por tanto, al igual que para el color azul-gris, que existe sobresegmentación para este pigmento. Esto se comprueba en la Figura 4-23.

Por otra parte, al evaluar el valor obtenido para el parámetro de Sensibilidad (Sen), se puede percibir que el valor conseguido, de la misma manera que para el color azul-gris y marrón oscuro, vuelve a ser bajo, siendo un 0,488. Por lo que la interpretación que se puede sacar de estos resultados es que, solo hay un 48,8% de que la segmentación realizada por el algoritmo sea real y por tanto, aproximadamente,

casi la mitad de las anomalías serán detectadas, señalando claramente la existencia del fenómeno de subsegmentación para el color marrón claro. La Figura 4-24 ilustra dicho fenómeno.

Finalmente, el valor obtenido por el parámetro Exactitud (Acc), es bastante alto, aunque un poco más bajo que en el caso del azul-gris, e indica que la precisión que presenta el algoritmo al realizar la segmentación es muy elevada.

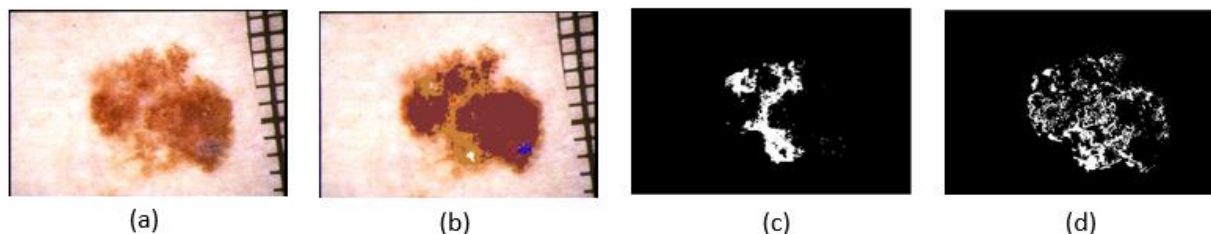


Figura 4-23. Sobresegmentación del color marrón claro en la lesión nfl072 perteneciente a clase Melanoma de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color marrón claro de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color marrón claro realizada por el algoritmo.

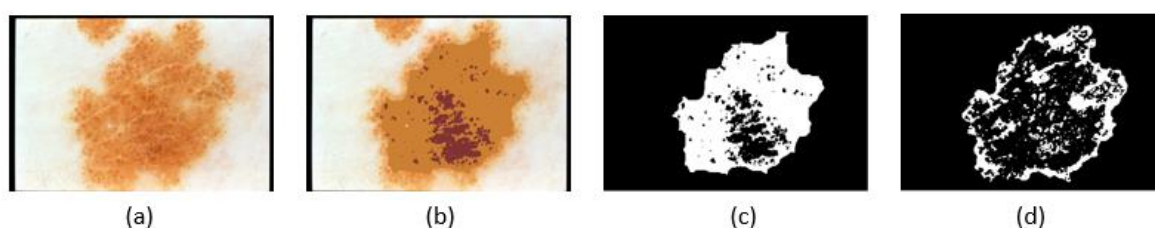


Figura 4-24. Subsegmentación del color marrón claro en la lesión Acl283 perteneciente a clase Clark nevus de la base de datos. Conjunto de imágenes: (a) Imagen original, (b) Versión clasificada de (a) por el dermatólogo, (c) Máscara de segmentación del color marrón claro de (b) realizada por el dermatólogo, (d) Máscara de segmentación del color marrón claro realizada por el algoritmo.

• Rojo

El análisis de este color no se ha podido realizar en las mismas condiciones que los anteriores, ya que se trata de un pigmento que solo ha sido capaz de ser comparado en tres ocasiones aunque haya sido detectado en un total de 6 veces por el algoritmo. Este inconveniente se ha debido a que cuando el algoritmo detectaba el color rojo, el especialista no lo había identificado y viceversa. Por tanto, solo las tres imágenes que se presentan en la Figura 4-25 han podido ser correctamente comparadas y evaluadas.

Una vez comentado todo lo anterior, hay que decir que al solo haber analizado tres imágenes que contienen el color rojo, los resultados pueden ser un poco imprecisos sobre el análisis del mismo por parte del algoritmo, pero dado que es lo que se ha obtenido se procederá a la examinación de dichos valores para los parámetros que miden la calidad de la segmentación del mismo.

Para empezar, el valor obtenido por el primero de los parámetros, VPP, apunta a que, a pesar de que no sea tan bajo como el del negro o marrón oscuro, existe posibilidad de sobresegmentación en las lesiones segmentadas. Esto se puede observar en la Figura 4-25, en concreto en la imagen (i), donde la lesión obtenida de forma automática ocupa más regiones que la imagen de referencia. En las otras imágenes también se puede apreciar este fenómeno pero de forma más sutil.

Seguidamente, el número dado por el parámetro de Sensibilidad (Sen), indica, como en el caso del color negro, que al ser un número bastante alto, la subsegmentación es un fenómeno descartado para este color. Además, se puede llegar a la conclusión de que una gran parte de las anomalías que se den durante la segmentación serán correctamente detectadas, ya que el valor del parámetro analizado está muy cerca de la unidad.

Con respecto al valor conseguido para el parámetro de Exactitud (Acc), como se muestra en la Tabla 4-1, se puede decir que es el valor más alto obtenido entre todos los colores evaluados en este trabajo. Al alcanzar este parámetro un valor tan elevado, se consigue que la precisión de la segmentación sea casi perfecta. Esto queda reflejado en la Figura 4-25, donde se puede apreciar la calidad de la segmentación realizada de forma automática.

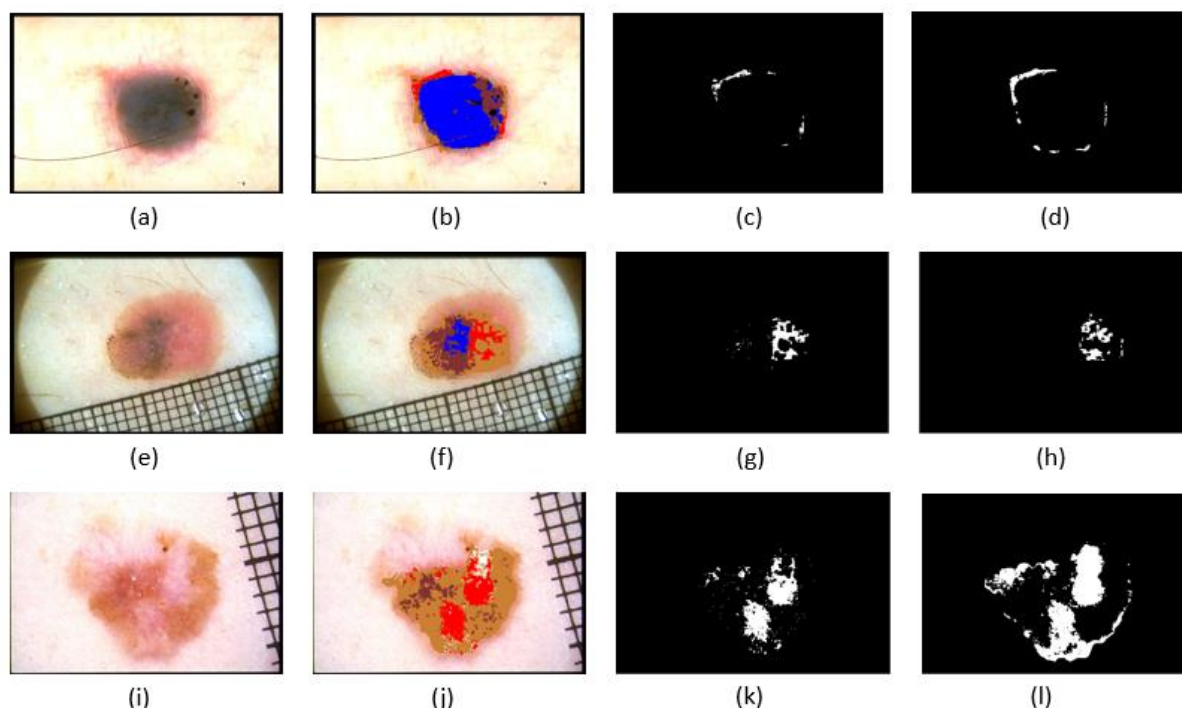


Figura 4-25. Imágenes (Nbl042, Ngl041, nal095, de arriba hacia abajo) analizadas para el color rojo pertenecientes a la clase Melanoma de la base de datos. Conjunto de imágenes: (a),(e),(i) Imágenes originales,(b),(f),(j) Versiones clasificadas de (a),(e),(i) ,respectivamente, por el dermatólogo, (c),(g),(k) Máscaras de segmentación del color rojo de (b),(f),(j) realizada por el dermatólogo, (d) ,(h),(l) Máscaras de segmentación del color rojo realizada por el algoritmo.

- **Blanco**

El color blanco, desgraciadamente, no ha podido ser evaluado ya que, a pesar de ser detectado por el especialista en varias ocasiones, como se puede observar en las Tablas 1 y 2 del Anexo B: Datos Relevantes, el algoritmo no ha conseguido identificar dicho color en ninguna de las lesiones analizadas pertenecientes a la base de datos. Quizás esto ha podido ser debido a que las muestras extraídas para el entrenamiento no han sido capaces de crear un modelo de mezcla que permitiera al algoritmo detectar el color blanco.

Debido a esta falta de valores para los parámetros que miden la calidad de segmentación del color blanco, los resultados totales que evalúan la eficiencia del algoritmo presentado se han podido ver afectados de una manera u otra, induciendo así un error al análisis de los mismos.

- **Resultados totales**

Antes de comentar los resultados finales, es importante destacar que los valores totales conseguidos para cada uno de los tres parámetros expuestos en la Tabla 4-1 se obtuvieron mediante la evaluación de las cantidades totales de TP, TN, FP, FN, por lo que la fila de resultados totales no representa la media de los valores que existen en la Tabla 4-1.

Una vez mencionado el detalle anterior, se pasa a la explicación de los resultados finales para cada uno de los parámetros de calidad. En primer lugar, se obtiene, para el parámetro VPP, un valor de 0,2984, que como se ha visto durante la explicación de los colores dermatoscópicos, es un número bastante bajo y que por tanto indica que el algoritmo, que en este trabajo se expone, presenta el fenómeno de sobresegmentación. Dicho fenómeno, constituye un tipo de error que determina que la segmentación no se está realizando de forma correcta. No obstante, este tipo de error puede mejorarse a base de empeorar otro, por ejemplo la subsegmentación, mediante el ajuste de los parámetros que el algoritmo contiene.

A continuación se analiza el segundo de los parámetros, la Sensibilidad (Sen). El valor total conseguido para dicho parámetro muestra que, aunque no es un número tan bajo como en el caso del marrón oscuro, sigue interpretándose como que el algoritmo presenta características del fenómeno de subsegmentación constituyendo así, al igual que en el caso de VPP, un error que da pie a confirmar que solo el 44,06% de las anomalías son detectadas, dato bastante bajo para ser considerado parte de una segmentación correcta.

Finalmente, el valor que presenta el tercer parámetro, la Exactitud (Acc), es bastante positivo, ya que el número obtenido es muy alto, en términos porcentuales se habla de aproximadamente un 90%, lo que, como se ha encontrado en análisis anteriores, indica que la precisión en la segmentación realizada es muy buena. Por tanto, cabe resaltar que esta alta precisión proporcionada por el algoritmo, constituye una gran ventaja a la hora de utilizarlo, ya que al hacerlo se obtendrán resultados de muy alta calidad en todo lo referido a la segmentación.

4.3.2.4 Matriz de confusión

Una correcta detección de los colores presentes en las lesiones, es un paso bastante importante a la hora de evaluar el funcionamiento de un algoritmo. De la misma manera, es necesario que se analice si a los píxeles de dichas lesiones se les está asignando correctamente los colores dermatoscópicos. En esta subsección, se trata esa última idea. El proceso comienza con el análisis de las lesiones pigmentadas por el algoritmo propuesto, mediante el cual se obtienen máscaras de segmentación que indican donde están localizados los píxeles de un determinado color. Esto se ha visto anteriormente en imágenes como las que recoge la Figura 4-13. Posteriormente, un especialista evalúa la clasificación realizada por el método propuesto, y señala a continuación, con el color correcto, la zona que ha sido mal clasificada. Este hecho se ilustra en la Figura 4-26.

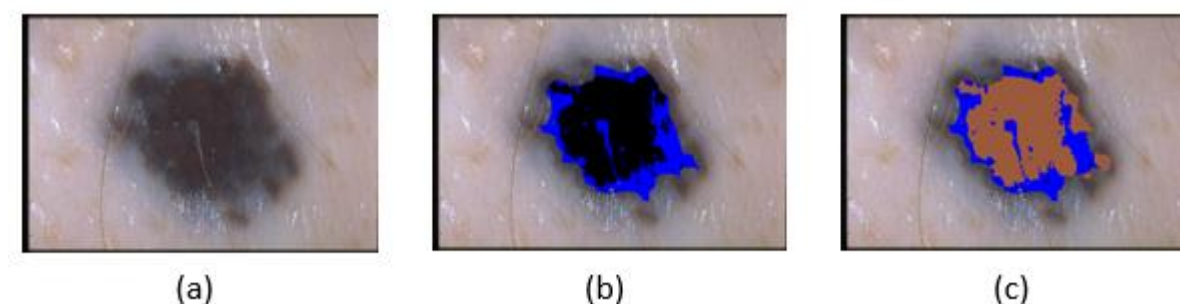


Figura 4-26. Evaluación de la imagen FBL032 de la clase Blue nevus de la base de datos. Conjunto de imágenes; (a) Imagen original, (b) Clasificación realizada por el algoritmo, (c) Clasificación realizada por el dermatólogo.

Con el fin de poder dar una idea general sobre la cantidad de píxeles, de cada tipo de color, que está mal clasificado, se procederá a la construcción de una tabla, Tabla 4-2, basada en la matriz de confusión, que sirve para el analizar qué pigmentos se confunden entre sí. Antes de evaluar los resultados ofrecidos por la Tabla 4-2, se explicará cómo se ha elaborado la misma, así como los procedimientos llevados a cabo para obtener los valores que la componen.

4.3.2.5 Construcción de la matriz de confusión

El objetivo que se quiere conseguir con la Tabla 4-2, es el de mostrar, con valores numéricos, la confusión de los colores de los píxeles que conforman las lesiones analizadas. Esta tabla se organiza de la siguiente forma: Las seis primeras columnas (con valores numéricos), muestran los colores que deberían haber sido identificados, mientras que las seis primeras filas (con valores numéricos), reflejan los colores que han sido detectados por el algoritmo. Finalmente, las celdas en las que columnas y filas coinciden, representan el porcentaje de área o píxeles erróneamente clasificados (que se han confundido), en otras palabras, cada celda muestra de forma aproximada, el porcentaje de área que ha sido identificada como un determinado color, pero realmente debería haber sido clasificada como otro.

En primer lugar, para poder obtener los valores que se muestran en la Tabla 4-2, se han comparado una a una las máscaras de colores detectadas por el algoritmo, con las máscaras de colores que deberían haberse detectado, las cuales han sido proporcionadas por el dermatólogo. En concreto, la mecánica seguida, como bien se ha explicado, ha consistido en cotejar todas las máscaras de los colores detectados por el algoritmo con todos aquellos que, para la lesión en cuestión, deberían haberse detectado realmente, obteniendo de cada una de esas comparaciones los píxeles resultantes de la intersección entre las dos segmentaciones cotejadas. Una vez obtenidos todos los píxeles para cada uno de los colores dermatoscópicos, se ha realizado la división de esa cantidad entre el total de los píxeles recogidos para cada tipo de color dermatoscópico evaluado por el especialista, dando lugar así a los diferentes valores conseguidos en la Tabla 4-2.

Por supuesto, no se ha realizado la comparación de colores con el mismo pigmento, en otras palabras, no se ha comparado la máscara del color del azul detectado por el algoritmo, con aquella, para ese mismo color, sacada por el especialista. Esto se debe, a que el propósito de esta tabla es evaluar la identificación de píxeles que deberían ser de un color, pero en la clasificación realizada por el algoritmo, han resultado ser de otro pigmento. Los valores totales e individuales de los píxeles de las máscaras cotejadas para todos los colores, se encuentran en la Tabla 3 del Anexo B: Datos Relevantes.

4.3.2.6 Análisis de los resultados Tabla 4-2

En este apartado se procederá al análisis de los valores expuestos en la Tabla 4-2, que se muestra a continuación. Es importante decir que la tabla que aquí se va analizar, se ha basado en la presentada en [4].

Tabla 4-2. Clasificación errónea de los colores de los píxeles de imágenes dermatoscópicas. Los valores presentes en esta tabla están en términos de porcentaje

Colores	Negro	Azul-Gris	Marrón Oscuro	Marrón Claro	Rojo	Blanco
Negro	97,72	48,22	63,95	5,39	26,12	0,10
Azul-Gris	1,22	48,10	4,46	12,57	17,14	88,82
Marrón Oscuro	0,14	0,45	25,39	49,15	29,66	0,31
Marrón Claro	0	0	2,67	23,96	0,36	0,76
Rojo	0	0	0,11	0,48	24,26	5,46
Blanco	0	0	0	0	0	0

En cuanto a los resultados que se presentan en la Tabla 4-2, se pueden extraer las siguientes conclusiones:

- Como se puede comprobar, los tres colores identificados con más porcentaje de error han sido el negro, el azul-gris y el marrón oscuro, en orden decreciente. Esta mala clasificación de los píxeles para dichos colores, puede deberse principalmente a que estos ocupen una mayor área en cada una de las lesiones, tal y como se muestra en la Figura 4-16 y 4-17, donde estos pigmentos son predominantes por encima de otros. Sin embargo, en cuanto al análisis del color negro, que es el que consigue el mayor porcentaje de error, hay que añadir que esta mala clasificación podría deberse también por el fenómeno de la sobresegmentación ya evaluado en la sección anterior. Hay que destacar además, que aunque el negro, azul-gris y marrón oscuro tengan los porcentajes más altos de error, sus porcentajes de no confusión (situados en la diagonal de la matriz (Tabla 4-2)), son los tres más elevados de la diagonal en orden también decreciente.
- Por otra parte, tal y como se observa en la Tabla 4-2, la celda que contiene el error más grande, de 0,8882, es la que representa el porcentaje de área que ha sido identificada como azul-gris pero debería haber sido clasificada como blanca. Esto puede ser uno de los factores del porqué el algoritmo presentado no es capaz de detectar el blanco, ya que lo confunde con azul-gris.
- Asimismo, analizando cuidadosamente las celdas de la Tabla 4-2, se puede decir que el color negro detectado por el algoritmo se clasifica de manera errónea para áreas de marrón oscuro y azul-gris, el color azul-gris para áreas principalmente blancas, el marrón oscuro para áreas en su mayoría de marrón claro, el marrón claro para píxeles clasificados como marrón oscuro, el rojo para zonas donde predomina el blanco y para el blanco, como se ha dicho con anterioridad, no se contemplan resultados. Dicha mala clasificación se puede deber en gran parte a la similitud entre esos colores en las lesiones analizadas. Además como se ha mencionado antes, fenómenos como la sobresegmentación o la subsegmentación, entre otros, han podido favorecer la clasificación errónea de los pigmentos de los píxeles para determinados colores.

4.3.3 Evaluación de la detección de colores en las lesiones

En esta subsección se hará un análisis de los colores detectados, por el algoritmo propuesto, en cada una de las lesiones que compone la base de datos de imagen. De esta manera, se evaluará si dichos colores han sido identificados correctamente, mediante la comparación de vectores (uno para cada color) que contienen los datos extraídos (unos y ceros) del parámetro **detected_colors** del algoritmo propuesto, con aquellos vectores creados

por el dermatólogo, obteniendo los TP, TN, FP, FN en sus versiones tanto totales como individuales (para cada imagen) para cada pigmento. Los colores detectados por el algoritmo, se proporcionan en la Tabla 1 del Anexo B: Datos Relevantes, así como los proporcionados por el médico que se muestran en la Tabla 4 del mismo anexo. En los Códigos 13 y 14 del Anexo A: Códigos, se puede observar el proceso de obtención de estos valores, implementado en lenguaje Matlab.

Una vez conseguidos los valores de las variables TP, TN, FP y FN (en sus formatos totales e individuales (En la Tabla 5 del Anexo B: Datos Relevantes se muestran)), se realiza, al igual que para el caso de 4.3.2, el cálculo de los parámetros encargados de medir la calidad de la detección de los colores: VPP, Sensibilidad (Sen) y Exactitud (Acc). Estos valores se muestran en la Tabla 4-3.

4.3.3.1 Análisis de los resultados Tabla 4-3

En la tabla que se muestra en este apartado, se pueden observar, en tanto por uno, los valores correspondientes a la detección de los colores en las lesiones de la piel.

Para poder entender mejor el significado de cada uno de los valores presentados en la Tabla 4-3, se analizarán y explicarán las interpretaciones de los parámetros de calidad de los colores por separado. Por último, se expondrá el alcance del algoritmo al poder extraer conclusiones de los valores finales de dichos parámetros de calidad. Es importante destacar antes de comenzar el análisis, que en Tabla 4-3, los colores evaluados presentan resultados mucho más favorables para los parámetros de calidad que los expuestos en la Tabla 4-2. Eso se debe a que es más fácil determinar qué colores existen en una lesión cutánea, que identificar con exactitud las regiones donde estos pigmentos se encuentran.

Tabla 4-3. VPP, Sen, Acc de la detección de los colores dermatoscópicos de las 127 imágenes en tanto por uno.

Colores	VPP	Sen	Acc
Negro	0,6147	1	0,6667
Azul-Gris	0,7640	0,85	0,7381
Marrón Oscuro	0,9556	0,8113	0,8095
Marrón Claro	0,9375	0,3158	0,4683
Rojo	0,5	0,3750	0,9365
Blanco	-	0	0,9206
Total	0,7791	0,6940	0,7566

A continuación se llevará a cabo el análisis anteriormente mencionado:

- **Negro**

El color negro recogido en la Tabla 4-3, presenta para el caso del parámetro VPP un valor mucho más alto que el obtenido en la Tabla 4-2. Esto puede indicar que para la detección de este pigmento el algoritmo se comporta de manera más eficiente que en la Tabla 4-2.

En el caso de la Sensibilidad (Sen) se alcanza el valor máximo posible para este parámetro, lo que indica que no se detectan FN para el color negro y que la clasificación del color negro se hace de forma perfecta.

En cuanto al parámetro de Exactitud (Acc), se puede percibir un decremento considerable respecto al caso de la Tabla 4-2. Eso se debe a que los parámetros anteriores han incrementado su valor, haciendo que el de la Acc disminuya en consecuencia. Además, esta disminución puede indicar que no se está enmascarando un mal etiquetado del color negro dentro de la lesión.

- **Azul-Gris**

En cuanto al color azul-gris recogido en la Tabla 4-3, presenta, al igual que el negro, un valor mucho más alto para VPP que el obtenido en la Tabla 4-2, incluso este valor es bastante más elevado que el que se consigue para el color negro, indicando así la buena eficiencia del algoritmo para dicho pigmento.

En el caso de la Sensibilidad (Sen) se alcanza un valor, en términos de porcentaje, de un 85%, superando al conseguido en la Tabla 4-2.

En cuanto al parámetro de Exactitud (Acc), se puede percibir un decremento considerable respecto al caso de la Tabla 4-2. Eso se debe, al igual que pasaba con el color negro, a que los parámetros anteriores han incrementado su valor, haciendo que el de la Acc disminuya en consecuencia. Además, esta disminución puede indicar que no se está enmascarando un mal etiquetado del color azul-gris dentro de la lesión.

- **Marrón Oscuro**

El color marrón oscuro recogido en la Tabla 4-3, presenta para el caso del parámetro VPP el valor más alto obtenido en dicha tabla (VPP = 95,56%). Este número es sin duda mucho más elevado que el conseguido en la Tabla 4-2, e indica que el algoritmo realmente es muy eficiente cuando detecta el marrón oscuro (para este color no hay apenas presencia de FP).

En el caso de la Sensibilidad (Sen), se alcanza el valor sutilmente menor que para el color Azul-Gris, pero aun así supera al que se obtiene en la Tabla 4-2.

En cuanto al parámetro de Exactitud (Acc), se puede percibir un decremento leve respecto al caso de la Tabla 4-2. Eso se debe, al igual que pasaba con los pigmentos anteriores, a que los dos primeros parámetros de calidad han incrementado su valor, haciendo que el de la Acc disminuya en consecuencia. Además, como los otros colores ya evaluados, esta disminución puede indicar que no se está enmascarando un mal etiquetado del color azul-gris dentro de la lesión.

- **Marrón Claro**

En lo referido al color marrón claro recogido en la Tabla 4-3, se puede observar que presenta, para el caso del parámetro VPP, un valor mucho más alto que el que se obtiene para este mismo parámetro en la Tabla 4-2. Este número tan elevado, del 93,75 % en términos de porcentaje, indica que el algoritmo en realidad es bastante eficiente en la detección del marrón claro, obteniendo muy poca cantidad de FP.

En el caso de la Sensibilidad (Sen), a diferencia de los colores analizados en párrafos superiores, se alcanza un valor muy bajo (el peor valor obtenido después del calculado para el color blanco), siendo incluso inferior que el conseguido en la Tabla 4-2 para este mismo color. Este hecho da a entender que el número de FN es más elevado que el observado para los colores ya analizados de la Tabla 4-3, y por tanto la Sensibilidad es más pequeña en el marrón claro que en cualquiera de ellos.

En cuanto al parámetro de Exactitud (Acc), se puede percibir un decremento bastante importante respecto al caso de la Tabla 4-2. Esto se debe en primera instancia, a que la Sensibilidad ha disminuido pero VPP ha mantenido un valor elevado, lo que se traduce en una disminución de este último parámetro.

- **Rojo**

El color rojo recogido en la Tabla 4-3, presenta para el caso del parámetro VPP un valor que mejora en un 10% respecto al obtenido en la Tabla 4-2. Esto indica que la eficacia real del algoritmo al evaluar dicho pigmento, no es demasiado buena.

En el caso de la Sensibilidad (Sen), se vuelve a alcanzar un valor demasiado bajo, y al igual que para el marrón claro, el número obtenido en la Tabla 4-3 es muy inferior al presentado en la Tabla 4-2. Todo esto implica que el valor de FN debe ser muy elevado para este color. 8

En cuanto al parámetro de Exactitud (Acc), se puede percibir un aumento bastante importante respecto a las cifras conseguidas para este parámetro en pigmentos anteriores. Dicho incremento puede deberse a la disminución de los valores de los dos parámetros ya analizados, intentando así enmascarar un fallo en la identificación del color rojo en las lesiones producidas en la piel.

- **Blanco**

Para el caso especial del color blanco, que como se ha dicho en apartados superiores, no ha sido detectado por el algoritmo propuesto ni una sola vez, se contemplan valores para el parámetro de Sensibilidad (Sen), siendo este de 0 e indicando claramente que existe un valor muy alto para la variable FN, tal y como se muestra en la Tabla 5 del Anexo B: Datos Relevantes, y para el parámetro de Exactitud (Acc), que consigue un valor muy alto, enmascarando así el error del algoritmo al ser incapaz de detectar el blanco en las imágenes dermatoscópicas.

- **Resultados totales**

Para concluir este apartado, se hace una evaluación de los resultados totales obtenidos para cada uno de los tres parámetros analizados.

En primer lugar el valor final obtenido para el primero de los parámetros (VPP = 77,91 %), indica que el algoritmo realmente se comporta bastante mejor que el caso expuesto en la Tabla 4-2, siendo así relativamente eficiente en cuanto a la detección de los pigmentos.

En cuanto a la Sensibilidad (Sen = 69,40%), se puede decir que aunque supera con creces el valor que presenta la Tabla 4-2, no se puede considerar una cifra lo suficientemente buena.

Finalmente, respecto a la Exactitud (Acc = 75,66%), se puede decir que presenta un valor medianamente elevado pero no en exceso, es decir, al mantenerse las cifras obtenidas para VPP y Sen en valores altos, la Acc no disminuye indicando que la precisión en la detección de los colores es normal y no trata de confundir, como en el caso del blanco en el que enmascara el mal rendimiento del algoritmo para la detección de dicho pigmento en las lesiones cutáneas, o en el caso del valor final de la Tabla 4-2 para Acc.

4.3.4 Comparación del algoritmo propuesto con otro método

En esta subsección se describe un método de detección estadística de los colores en imágenes dermatoscópicas mediante una estimación de probabilidades basada en texturas, que emplea la misma base de datos de imagen que utiliza el algoritmo propuesto en el capítulo 3, así como la misma evaluación hecha por el dermatólogo. Debido a ello, se realizará una comparación de los valores finales obtenidos para los parámetros de calidad de la Tabla 4-3 con los que se obtienen de la misma manera para la técnica propuesta en [4].

4.3.4.1 Descripción del funcionamiento de la técnica

Actualmente los métodos existentes encargados del diagnóstico de lesiones cutáneas, se apoyan, en gran medida, en los colores que se detectan dentro de las mismas. Este hecho, deja entre ver, en términos de diagnosis, la importancia que tiene el color en la dermatoscopia. Según [4], se sabe que en las lesiones malignas se pueden encontrar más de tres colores (entre ellos destacan el negro, el azul-gris, el rojo y el blanco que se producen con más frecuencia en las lesiones cancerosas), mientras que en las benignas los colores detectados se sitúan por debajo de tres o como máximo llegan a esa cifra. La técnica que se propone en esta última sección, tiene como principal objetivo la identificación de forma automática de los colores presentes en lesiones de la piel. Esta detección de colores, se ha llevado a cabo mediante la maximización de la probabilidad a posteriori de un píxel, de pertenecer a alguno de los seis colores dermatoscópicos, una vez dados tanto su valor de color como los valores de colores de los píxeles vecinos [4]. Por otra parte, para que las probabilidades anteriores sean calculadas, es necesaria la estimación de los diferentes términos implicados en esos cálculos. Para poder realizar dichas estimaciones, se llevan a cabo dos evaluaciones de una base de datos (EDRA) compuesta por 200 imágenes dermatoscópicas [4]. En primer lugar, la primera de las evaluaciones se encarga de comprobar si todos los colores identificados dentro de una lesión realmente están presentes en ella. Por el contrario, la segunda evaluación analiza si a cada uno de los píxeles que forman la lesión, se les ha asignado el color correcto. Ambas evaluaciones se verán reflejadas mediante parámetros como la Sensibilidad, Exactitud o VPP, en el caso de la primera, y de porcentajes que representen la mala clasificación de los colores en el caso de la segunda. Además de todo lo mencionado, es importante comentar que para la identificación de colores, en concreto de los 6 colores dermatoscópicos, no se usa tan solo la información de color obtenida de las lesiones pigmentadas, sino que también se emplea la información de la textura de las mismas. En la Figura 4-27, se muestra un ejemplo de la

detección de los colores en lesiones completas, realizadas por la técnica presentada, mientras que en la Figura 6-8, se puede observar el porcentaje de área que ha sido clasificada de forma incorrecta.

Cómo se ha explicado en el párrafo anterior, la metodología seguida por este algoritmo es la siguiente: En primer lugar se formula el problema del máximo a posteriori, a través del cual se estima la probabilidad deseada, en otras palabras, la probabilidad de que un píxel pertenezca a cada uno de los 6 colores dermatoscópicos (6 probabilidades), dado su valor de color, que se trata de un vector de 3 coordenadas, así como los valores de los píxeles adyacentes. Seguidamente, se pasa al cálculo de dichas probabilidades. Para ello, se crean diferentes modelos de distribución de probabilidades a partir de las imágenes de entrenamiento presentadas, que al igual que en el capítulo 3, consisten en pequeños parches que muestran los seis posibles colores [4]. Finalmente, se expone el procedimiento por el cual se clasifican los píxeles como colores. Este proceso consiste en que, para cada uno de los píxeles presentes en las imágenes de prueba, la probabilidad de pertenecer a cada uno de los 6 colores dermatoscópicos es calculada (a partir de las funciones de distribución que se comentaban ante). Una vez calculadas, la etiqueta de color que haya obtenido la mayor probabilidad será asignada al píxel analizado [4]. Esto se hace de forma simultánea para todos los píxeles que componen la lesión, por lo que una vez que se termine este análisis, el proceso de clasificación estará completo.

Los resultados que se obtienen al aplicar este algoritmo, muestran que este método es capaz de detectar correctamente los colores presentes en una lesión cutánea, lo que supone una ventaja en cuanto a la diagnosis de melanomas. Además, como se ha podido ver en la Figura 4-28, este algoritmo es capaz de determinar para cada región de color, que porcentaje de la misma es errónea y cual debería haber sido su color original. Por otra parte, hay que destacar que la información de textura extraídas de las lesiones, aportan a este método un carácter de superioridad frente a otros, como el propuesto en el capítulo 3, que solo utilizan información de color. El uso de características de textura, por tanto, es un factor sumamente importante para conseguir una correcta clasificación de los colores existentes en las lesiones de la piel. Por último, los valores totales obtenidos para los parámetros de VPP (VPP=89%), Sensibilidad (Sen=89%) y Exactitud (Acc=90%), indican la buena calidad del análisis que hace este algoritmo, actuando de una manera correcta superando a otros métodos, como el presentado en este trabajo, cuyos valores de VPP y Sensibilidad están muy por debajo de los obtenidos con esta técnica.

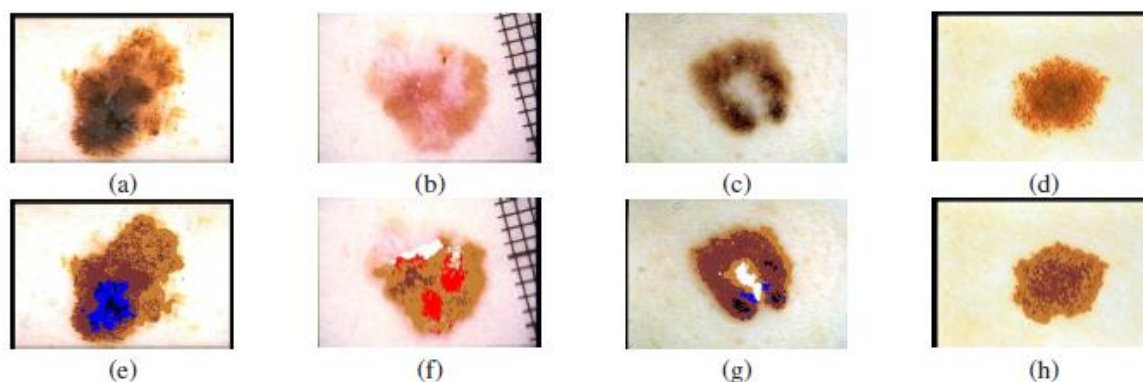


Figura 4-27. Ejemplos de la detección de colores en lesiones completas. Conjunto de imágenes: (a),(b),(c),(d) Imágenes originales, (e),(f),(g),(h) Versiones clasificadas realizadas por el algoritmo.

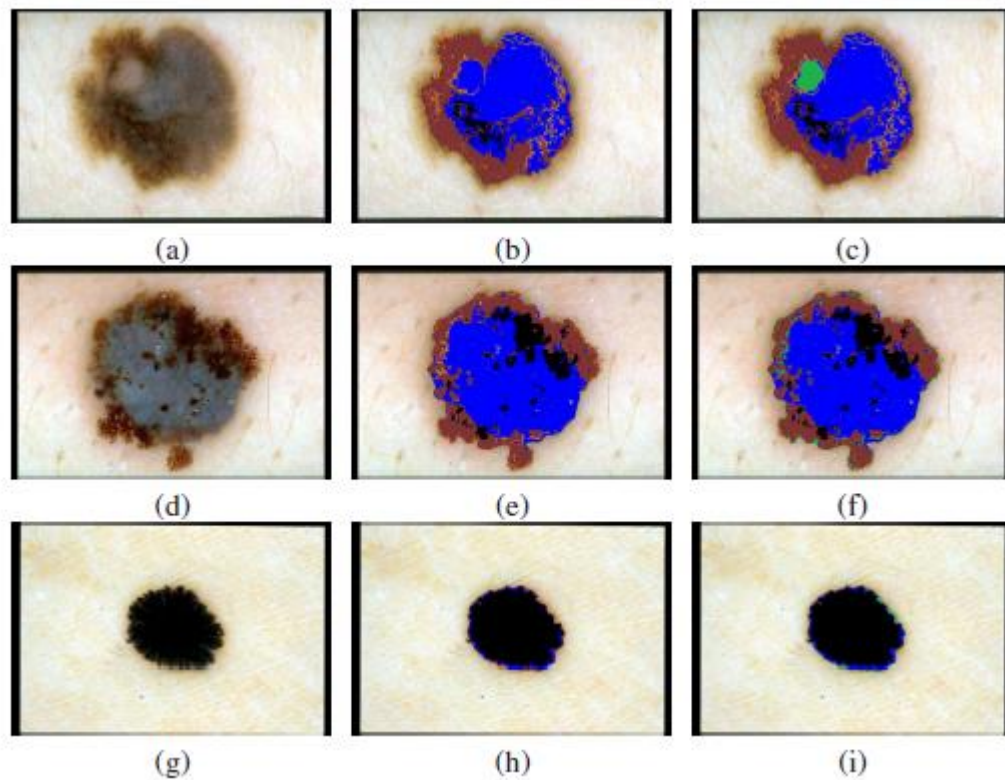


Figura 4-28. Ejemplos de la evaluación del dermatólogo: (a) Imagen original, (b) Detección de color de (a) realizada por el algoritmo, (c) Imagen evaluada por el dermatólogo en la que las zonas verdes simbolizan las regiones mal clasificadas (esa región debería haber sido marrón clara en vez de azul). (d) Imagen original, (e) Detección de color de (d) realizada por el algoritmo, (f) Imagen evaluada por el dermatólogo en la que las zonas verdes simbolizan las regiones mal clasificadas (de acuerdo con el dermatólogo no hay marrón claro). (g) Imagen original, (h) Detección de color de (g) realizada por el algoritmo, (i) Imagen evaluada por el dermatólogo en la que las zonas verdes simbolizan las regiones mal clasificadas (esa región mal clasificada corresponde a piel saludable que se ha incluido en la lesión debido a la segmentación realizada por el algoritmo) [4]

5 CONCLUSIONES

Como se ha comentado en capítulos anteriores, en este trabajo se expone el funcionamiento de un algoritmo capaz de identificar colores, concretamente seis (negro, azul-gris, marrón oscuro, marrón claro, rojo y blanco), en imágenes dermatoscópicas recogidas en una base de datos, mediante el empleo de modelos de mezcla Gaussiana. Asimismo, uno de los objetivos de la implementación de dichos modelos, es el que implica que puedan ser utilizados en sistemas CAD capaces de reproducir, casi a la perfección, los pasos seguidos por un especialista en la detección de esos seis colores presentes en lesiones pigmentadas. Ahora bien, los resultados experimentales que en el capítulo cuatro se exponen, dejan entre ver una serie de conclusiones que serán debidamente analizadas a continuación.

5.1 Conclusiones extraídas de los resultados experimentales

En primer lugar, los resultados que se refieren a la asignación de los colores dermatoscópicos a los píxeles presentes en las lesiones evaluadas, exponen que el algoritmo propuesto presenta fenómenos como el de sobresegmentación, que indica que los píxeles que segmenta el algoritmo, solo se corresponden realmente en un porcentaje muy pequeño (según la Tabla 4-1, en un 29,84%) con los pertenecientes a la región que segmenta el especialista. De la misma manera, también destaca el efecto de la subsegmentación, que señala que un porcentaje bajo, en torno al 44%, de la segmentación real ha sido segmentada por el algoritmo, dando así a entender, la poca eficiencia que presenta el mismo, ya que menos de la mitad de las anomalías presentes en las lesiones serían detectadas. Por lo tanto, desde el punto de vista de los parámetros VPP y Sen, encargados de medir los dos fenómenos anteriores, la técnica que se presenta en este documento no es demasiado útil debido a su bajo VPP y a su valor intermedio para Sen. Por el contrario el parámetro de Exactitud (Acc) toma un valor muy elevado (Acc = 90,03 %) debido a la existencia de una cantidad pequeña de píxeles mal etiquetados en relación al tamaño total de la imagen implica la obtención de una elevada precisión, enmascarando así los errores cometidos por el algoritmo. No obstante, el parámetro de Exactitud puede ser muy grande si, como en casos parecidos a este, se obtienen valores de VPP y Sen muy pequeños.

En cuanto a los resultados extraídos de la Tabla 4-2, se puede observar que los colores que presentan mayor porcentaje de ser clasificados erróneamente, son el negro, azul-gris y marrón oscuro, los cuales suelen estar presentes en la mayoría de las lesiones, ocupando un porcentaje bastante alto de área, de ahí su alto índice de error. De la misma manera, en el caso del color negro y el marrón oscuro, destacan fenómenos como la sobresegmentación, y en el caso del marrón oscuro, la subsegmentación, los cuales constituyen una fuente de error a considerar. Asimismo, el pigmento blanco presente en las lesiones, ha sido identificado por el algoritmo como azul-gris, pudiendo esto indicar la causa de la no detección por parte del método planteado del color blanco, a parte de su poca frecuencia de aparición en las lesiones evaluadas. Esta confusión ha podido deberse, posiblemente, a los parches utilizados para el entrenamiento del algoritmo, entre otros factores. Por último, hay que decir también que la similitud entre los colores presentes en las lesiones pigmentadas ha podido ser otro de los problemas que hayan causado esa mala clasificación de los pigmentos.

Asimismo, los resultados que se refieren a la detección y clasificación de los colores dermatoscópicos presentes en las lesiones evaluadas, y que se muestran en la Tabla 4-3, dan a entender que para el caso de la detección de colores, los valores conseguidos para los parámetros de VPP (VPP = 77,91%), Sensibilidad (Sen = 69,40%) y Exactitud (Acc = 75,66%) aumentan considerablemente respecto de aquellos obtenidos en el caso de la asignación de los colores dermatoscópicos a los píxeles presentes en las lesiones analizadas. Esto lleva a la conclusión de que es mucho más difícil la coincidencia, entre el algoritmo planteado y el especialista, de las regiones de color identificadas, que de la detección de la existencia de los diferentes colores (solo se indica si están presentes o no mediante unos y ceros respectivamente) en las lesiones evaluadas, ya que al comparar regiones de color, se cotejan gran cantidad de píxeles a la vez para un solo color, mientras que al comparar colores detectados solo se coteja una cifra para un color determinado.

Por otra parte, una vez analizados los resultados anteriores, se puede apreciar que para ambas tablas, la presencia de colores tales como el blanco y el rojo, no presenta la misma frecuencia de aparición que el resto de colores dermatoscópicos. Además de ello, las áreas en las que estos colores son detectados no son muy extensas (más bien son de tamaño pequeño). Todos estos efectos, denotan la existencia de un problema de desequilibrio en las muestras evaluadas. En consecuencia, cualquier error que se dé en las lesiones en las que aparezcan tanto el color rojo como el blanco, será mucho más importante, desde el punto de vista de las métricas de evaluación propuestas, que cualquier otro que se dé para los colores restantes. Igualmente, se ha podido comprobar que el algoritmo es capaz de segmentar regiones mayores en las lesiones evaluadas, a diferencia de la segmentación realizada por el especialista, que se caracteriza por la extracción de áreas menores. En otras palabras, esto significa que el método descrito es capaz de evaluar zonas que no necesariamente forman parte de la lesión, pero que al estar pigmentadas, pueden suponer una fuente de error a tener en cuenta.

De la misma manera, aunque este método de detección y clasificación automáticas haya supuesto una ventaja en lo que se refiere a velocidad, como puede apreciarse en el capítulo tres del trabajo, todavía sigue siendo un algoritmo relativamente lento en su ejecución, si bien es más rápido que aquellos que solo implementan la técnica EM. El proceso automático expuesto en este documento suele tener una duración media de entre unos 5 a 15 minutos, consolidándose así como un algoritmo que no es inmediato, lo que dificulta la agilidad a la hora de recoger los resultados que miden su eficiencia. No obstante, sigue siendo una mejora respecto a aquellos que solo se basaban en técnicas EM.

Otra de las causas de error existentes al analizar el comportamiento del algoritmo, tiene que ver principalmente con la extracción de imágenes de la base de datos. Este problema surge, dado que, aunque todas las lesiones evaluadas fueron sacadas de la misma base de datos (Interactive Atlas of Dermoscopy o Atlas Interactivo de la Dermatoscopia), estas se obtuvieron de tres hospitales diferentes [4], y por tanto expuestas a condiciones de luz diferentes. Asimismo, muchas de estas imágenes no han sufrido una etapa de preprocesado, en la cual se retiran tanto burbujas de gel como pelos, lo que supondría una mejora a la hora de realizar la detección de los colores.

Además de ello, también se sabe que el algoritmo presentado, como se ha comentado anteriormente, no realiza correctamente la detección del color blanco. Esto se debe, entre otros factores, a que según el criterio clínico, el color blanco se considera detectado cuando los píxeles que forman la lesión son mucho más brillantes o luminosos que la piel saludable que rodea la lesión [4]. Por lo que, si el algoritmo considerara la información del color de la piel adyacente en la muestra, mejoraría esta identificación del blanco.

6 MEJORAS Y LÍNEAS FUTURAS

Como se ha podido observar en el capítulo anterior, y en vista de los resultados analizados, el algoritmo presenta más inconvenientes que ventajas cuando se hace uso del mismo. Por ello, en esta parte del trabajo se recogerán varias soluciones que podrían suponer una mejora para el funcionamiento de dicho algoritmo si a él se le aplicasen. Además se describirá el funcionamiento de aquellos métodos que implementan dichas mejoras. A continuación se expondrán varios de estos sistemas con sus pros y contras.

6.1 Mejoras del algoritmo propuesto

En esta sección se presenta una mejora para el método presentado en el capítulo 4.

6.1.1 Empleo de la constancia del color para un mejor análisis de imágenes dermatoscópicas

Muchos de los métodos encargados de la detección de melanomas en imágenes dermatoscópicas, están sujetos a varias condiciones de adquisición. Esto quiere decir que, el rendimiento de dichas técnicas se ve afectado cuando el entorno de la adquisición de esas imágenes cambia, o cuando los datos proceden de múltiples fuentes [31], ya que dichos sucesos alteran el color de las imágenes. Esto es lo que sucede con la base de datos EDRA [31], la cual, al igual que la propuesta en el capítulo 4 del trabajo y como se ha comentado en el capítulo 5 del mismo, contiene imágenes procedentes de tres hospitales distintos, y por tanto con condiciones de iluminación completamente distintas. Por ello, los sistemas CAD, entre los que se incluyen el algoritmo propuesto, así como las técnicas mencionadas anteriormente, se ven muy perjudicados por ese cambio de iluminación, el cual introduce irregularidades en los valores de las características relacionadas con el color, tales como los histogramas de color o vectores de medias de color. Esto hace que dichos sistemas sean menos robustos y, por ende, más propensos a errores en el análisis de imágenes provenientes de varias fuentes [31]. El cerebro humano, por el contrario, no tiene dificultades cuando estos cambios ocurren, ya que es capaz de compensar la variabilidad de la luz en las muestras, lo que hace que los dermatólogos confíen más en sus habilidades que en este tipo de programas. Con objeto de solventar dicho problema, en esta sección se llevará a cabo un análisis de técnicas de compensación del color, que tienen como fin principal mitigar la influencia del entorno de adquisición sobre las características de color extraídas de las lesiones. En la Figura 6-1, se pueden observar algunos ejemplos de imágenes conseguidas bajo condiciones de iluminación distintas.

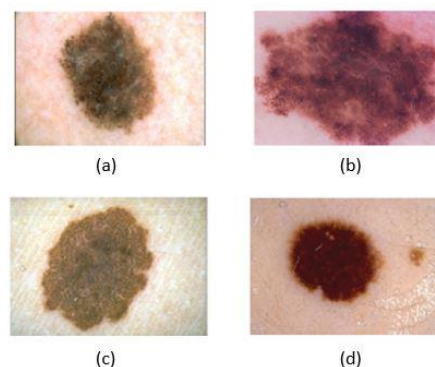


Figura 6-1. Ejemplos de imágenes obtenidas bajo diferentes condiciones de iluminación. Conjunto de imágenes:

(a) y (b) son melanomas, (c) y (d) son lesiones benignas

Uno de los métodos principales que ayuda a reducir el problema de la normalización de color y es capaz de tratar con datos de múltiples fuentes e incluso de procedencia desconocida, es el llamado método de Constancia del Color. A continuación se va a proceder a la explicación de esta técnica.

En primer lugar, hay que decir que el principal objetivo de los métodos que se apoyan en la constancia del color, es el de transformar los colores presentes en una imagen, que por supuesto es de orígenes desconocidos (referido a la iluminación de la misma), para que estos sean idénticos a los conseguidos en imágenes expuestas a una iluminación ideal [31]. Con el fin de conseguir dicha tarea, primeramente se estima el color de la fuente de luz en coordenadas RGB ($\mathbf{e} = [e_R e_G e_B]^T$) y posteriormente se transforma la imagen usando la iluminación estimada. Una gran diversidad de algoritmos se han utilizado para llevar a cabo esta estimación del color de la fuente de luz, pero aquí se ha optado por uno de ellos en concreto: el método de Shades of Gray [32], empleado para la aplicación de la constancia del color. Dicha técnica, no requiere del conocimiento de las propiedades de adquisición del sistema, ni siquiera necesita de una fase de entrenamiento, ella misma se encarga de proporcionar una fase de calibración. Como se menciona en [31], a pesar de ser un método bastante simple, la eficiencia del mismo es comparable al rendimiento de algoritmos más complejos basados en entrenamiento, y además es más rápido y fácil de implementar. Cuando el proceso de la estimación del origen de luz es completado, se pasa a la transformación de la imagen. Para realizar esta conversión, se emplea un método bastante sencillo: el modelo diagonal de von Kris [31]. En la Figura 6-2 se muestran ejemplos de la constancia del color.

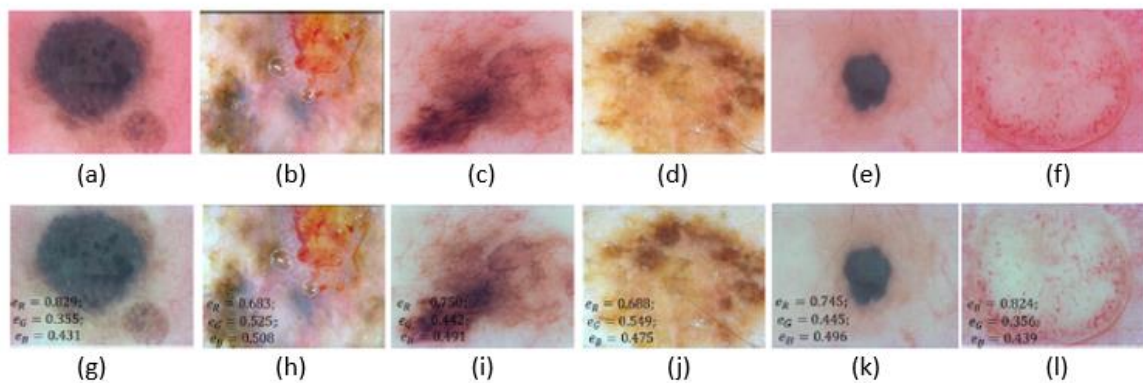


Figura 6-2. Ejemplos de constancia del color. Conjunto de imágenes: (a),(b),(c),(d),(e),(f) Imágenes originales, (g),(h),(i),(j),(k),(l), Imágenes obtenidas mediante Shade of Gray $p=6$. Las tres primeras columnas son melanomas, mientras que las tres últimas son lesiones benignas.

Finalmente, se evalúa la influencia de la normalización del color conseguida, en dos aplicaciones dermatoscópicas distintas, fuertemente dependientes de las características del color. La primera de ellas, es un sistema CAD que se encarga del diagnóstico de las lesiones producidas en la piel mediante el modelo BoF y los histogramas de color [29]. Por otra parte, la segunda aplicación se basa en la idea de que, para el desarrollo de un sistema CAD que identifique y cuente el número de colores presentes en imágenes dermatoscópicas, empleando mezclas Gaussianas (como el presentado en [1]), la constancia del color es muy relevante.

Los resultados obtenidos para la primera de las aplicaciones, han concluido que la constancia del color supone una mejoría en el rendimiento del modelo BoF en un 14% [31], en lo referido a la clasificación de imágenes provenientes de múltiples sitios. Sin embargo, cuando los datos proporcionados tienen como origen una sola fuente, no se aprecia ninguna mejora o degradación al utilizar métodos que impliquen el uso de la constancia del color [31]. Por otra parte, los resultados obtenidos para la segunda aplicación, basada en la detección de los colores, indican que se han conseguido mejoras importantes incluso cuando los datos provenían de un único origen. Por tanto, la corrección del color parece mejorar la discriminación del mismo [31]. También hay que destacar que, tanto en el espacio de color HSV como en el RGB, la constancia del color funciona muy bien, aportando avances en el rendimiento de los dos sistemas contemplados. En las Tablas 6-1 y 6-2 se muestran los valores conseguidos para los parámetros de Sensibilidad y Exactitud, ante la presencia de una única fuente y de

varias. Con dichas tablas, se respaldan las conclusiones relatadas en este párrafo, ya que como se ve en la Tabla 6-1, la aplicación de la constancia del color implica una mejora para el caso de fuentes distintas, mientras que en la Tabla 6-2, se observa como al utilizar una única fuente, la constancia del color no degrada la productividad del clasificador.

Tabla 6–1. Resultados de la clasificación para la base de datos EDRA (múltiples fuentes), con la corrección de la constancia del color y sin ella.

Constancia del color	Sensibilidad (Sen)	Exactitud (Acc)
No aplicada	71%	63,1%
Shades of Gray	79,7%	77,8%

Tabla 6–2. Resultados de la clasificación para la base de datos PH² (fuente única), con la corrección de la constancia del color y sin ella.

Constancia del color	Sensibilidad (Sen)	Exactitud (Acc)
No aplicada	92,5%	84,1%
Shades of Gray	92,5%	84,3%

6.2 Líneas Futuras: Aplicación de la detección de color para una posterior clasificación de la malignidad y benignidad de las lesiones.

En esta sección se hará un resumen de las técnicas futuras a las que se les puede aplicar la detección de color con el fin de obtener una clasificación de las lesiones pigmentadas analizadas, en benignas y malignas.

6.2.1 Empleo de un sistema inteligente de soporte a decisiones para la detección del cáncer de piel a partir de imágenes dermatoscópicas

Según [2], el desarrollo de un sistema robótico (e-Carers) capaz de dirigir, a largo plazo, mecanismos de monitorización de la salud de forma automática, así como detectar y diagnosticar enfermedades de forma eficiente, supone todo un reto. En esta subsección, se hace frente a retos como el anterior mediante la proposición de un sistema inteligente de soporte a decisiones, encargado la detección del cáncer cutáneo a partir de imágenes dermatoscópicas. Dicho sistema, por supuesto, podría formar parte en un futuro de ese robot inteligente de servicios que, como se comentaba anteriormente, se encarga del cuidado de la salud en el entorno del hogar con el fin de hacer diagnósticos de forma más rápida y poder tratar enfermedades con una mayor eficacia. La técnica expuesta en esta subsección, es desarrollada para identificar lesiones malignas y benignas producidas en la piel mediante el uso de varios pasos, tales como el pre-procesado, la eliminación del ruido, la segmentación o la extracción, selección y clasificación de características de determinadas regiones que forman las lesiones analizadas [2].

6.2.2 Empleo de una técnica de reconstrucción 3D para la clasificación automática de lesiones dermatoscópicas en la piel

El melanoma, al igual que se ha explicado en las otras técnicas, tiene unos índices de mortalidad muy elevados, situándose como uno de los cánceres de piel más agresivos. Este tipo de cáncer, es muy peligroso cuando crece más allá de la dermis, y por tanto, la profundidad es un factor clave para poder diagnosticar esta enfermedad [33]. En esta subsección, se propone un sistema dermatoscópico automático de carácter no invasivo, que tiene en cuenta la profundidad estimada de las lesiones producidas en la piel para el proceso de diagnóstico [33]. Dicho sistema, se trata de una técnica de reconstrucción 3-D de la lesión de la piel, el cual emplea para ello, la profundidad estimada obtenida a partir de imágenes dermatoscópicas [33]. Las características necesarias para realizar esta reconstrucción 3-D, y las cuales se extraen cuidadosamente, son la profundidad y la forma 3-D. Además de ellas, las características de color, textura y forma 2-D también son extraídas. Es importante destacar, que como se ha visto, el proceso de extracción de características es el punto clave para poder conseguir resultados precisos.

En conclusión, se puede decir que gracias al empleo de esta técnica es posible obtener la profundidad del melanoma, aspecto clave para conocer el estado del cáncer y así actuar contra él de manera consecuente. Esto se refleja en la Figura 6-3. Además, analizando los resultados obtenidos una vez aplicado el método, se puede extraer que la inclusión de las características de forma 3D, incluyendo por supuesto las que estiman la profundidad de la lesión, mejoran considerablemente el rendimiento del algoritmo y ayudan a realizar un diagnóstico mucho más preciso de varios tipos de lesiones cutáneas [33].

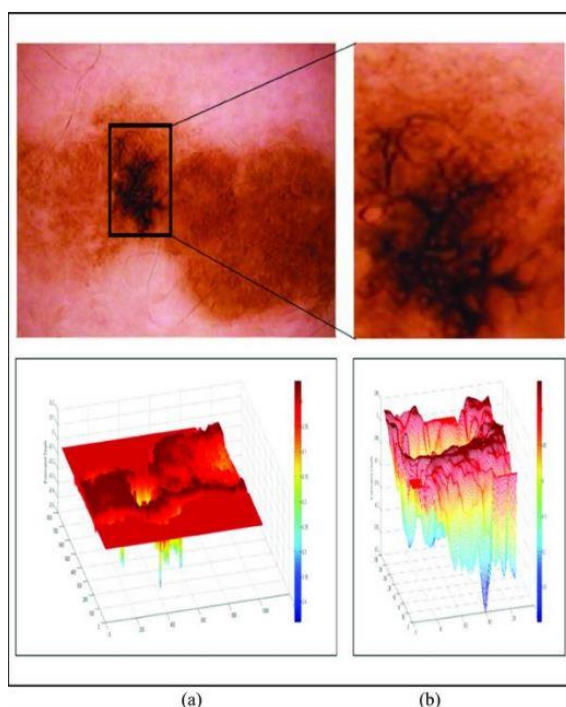


Figura 6-3. Imagen de seguimiento y proyecciones 3-D de la profundidad de un melanoma. Conjunto de imágenes: (a) Imagen de seguimiento (imagen superior) y profundidad estimada (imagen inferior), (b) Región de interés del melanoma (imagen superior) y profundidad estimada de la misma (imagen inferior) [33].

6.2.3 Empleo de una técnica de extracción de características de color local basada en superpixel para la clasificación de lesiones cutáneas

Debido a propiedades estructurales complejas, junto con la presencia de un poco de proximidad de color [34], las dos enfermedades, nevus común y melanoma, han supuesto un problema para los especialistas a la hora de diferenciar una de la otra, solamente usando la inspección visual. En esta sección, se expone un método para la identificación de ambas enfermedades, mediante una técnica de clasificación de las lesiones producidas en la piel, mediante ordenador. Dicha técnica, se encarga de la extracción de varias características tanto globales como locales, y una vez extraídas, se aplica un proceso de eliminación de características recursivas. Además de ello, este sistema emplea para la construcción de un descriptor de características de color local, encargado de la clasificación y almacenamiento de las mismas, varias regiones aisladas mediante la agrupación de píxeles, los cuales aportan información importante, formando regiones de superpíxeles [34].

Analizando los resultados que proporciona este método, se puede concluir que usando una técnica de extracción de características locales basadas en las regiones de superpíxeles, se ha podido localizar e identificar la información del color que está presente en ambos tipos de lesiones (melanoma y nevus común) y que hace que una sea diferente de la otra. No obstante, hay que mencionar que se ha utilizado un número de superpíxeles uniforme para todas las imágenes, aunque usualmente el número de superpíxeles real no sea uniforme y su valor óptimo dependa de la información de la presencia de color. La optimización del número de superpíxeles para una imagen dermatoscópica en particular, es algo que se espera conseguir en futuros trabajos [34].

6.2.4 Empleo de dos sistemas para la detección de melanomas en imágenes dermatoscópicas usando características de color y textura

En la actualidad, el melanoma es uno de los tipos de cáncer de piel con más índice de muertes registradas hasta la fecha. Por tanto, es de sentido común que se hayan desarrollado una gran cantidad de técnicas y métodos capaces de realizar diagnósticos mucho más eficientes y fiables. Una de las técnicas propuestas, casi a la misma vez que la de este trabajo, y que alcanza estos objetivos, es la que describe dos sistemas distintos empleados para la detección de melanomas en las imágenes dermatoscópicas [3]. El primero de ellos se encarga de la clasificación de las lesiones de la piel mediante el uso de métodos globales, mientras que el segundo emplea características locales así como clasificadores BoF (Bag of Features) [3]. El objetivo a alcanzar con la implementación de estos sistemas, a parte de la consolidación de uno de ellos como el que mejor clasifica las lesiones producidas en la piel, es el de comparar el papel que tienen tanto las características de color como las de textura, dentro de la clasificación de las lesiones y así determinar cuál de esos conjuntos de características es el más restrictivo. En otras palabras, se pretende realizar dos tipos de comparaciones, una encargada de comprobar que sistema hace mejor la clasificación, y otra que recoge el rendimiento de los dos conjuntos de características (color y textura) en la identificación de los melanomas, determinando, además, cuál de ellos es el mejor discriminante [3].

REFERENCIAS

- [1] Barata, C., Figueiredo, M. A. T., Celebi, M. E., & Marques, J. S. (2014). Color identification in dermoscopy images using Gaussian mixture models. In ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. <https://doi.org/10.1109/ICASSP.2014.6854274>
- [2] Tan, T. Y., Zhang, L., & Jiang, M. (2016). An intelligent decision support system for skin cancer detection from dermoscopic images. 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2194–2199. <https://doi.org/10.1109/FSKD.2016.7603521>
- [3] Barata, C., Ruela, M., Francisco, M., Mendonca, T., & Marques, J. S. (2014). Two systems for the detection of melanomas in dermoscopy images using texture and color features. IEEE Systems Journal. <https://doi.org/10.1109/JSYST.2013.2271540>
- [4] Saez, A., Acha, B., Serrano, A., & Serrano, C. (2018). Statistical detection of colors in dermoscopic images with a texton-based estimation of probabilities. IEEE Journal of Biomedical and Health Informatics, 2194(c), 1–1. <https://doi.org/10.1109/JBHI.2018.2823499>
- [5] Stolz, W., Riemann, A., Cognetta, A. B., Pillet, L., Abmayer, W., Holzel, D., ... Braun-Falco, O. (1994). ABCD rule of dermatoscopy: A new practical method for early recognition of malignant melanoma. European Journal of Dermatology.
- [6] Menzies, S. W., Ingvar, C., Crotty, K. A., & McCarthy, W. H. (1996). Frequency and morphologic characteristics of invasive melanomas lacking specific surface microscopic features. Archives of Dermatology. <https://doi.org/10.1001/archderm.132.10.1178>.
- [7] Argenziano, G., Fabbrocini, G., Carli, P., De Giorgi, V., Sammarco, E., & Delfino, M. (1998). Epiluminescence Microscopy for the Diagnosis of Doubtful Melanocytic Skin Lesions. Archives of Dermatology. <https://doi.org/10.1001/archderm.134.12.1563>.
- [8] Silva, C. S. P., Marcal, A. R. S., Pereira, M. A., Mendonça, T., & Rozeira, J. (2012). Separability analysis of color classes on dermoscopic images. in Proceedings of the 9th International Conference on ImageAnalysis and Recognition - Volume Part II, 2012, 268–277.
- [9] Silva, C. S. P., & Marcal, A. R. S. (2013). Colour-based dermoscopy classification of cutaneous lesions: An alternative approach. Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization. <https://doi.org/10.1080/21681163.2013.803683>
- [10] Seidenari, S., Pellacani, G., & Grana, C. (2003). Computer description of colours in dermoscopic melanocytic lesion images reproducing clinical assessment. British Journal of Dermatology. <https://doi.org/10.1046/j.1365-2133.2003.05496.x>
- [11] Pellacani, G., Grana, C., & Seidenari, S. (2004). Automated description of colours in polarized-light surface microscopy images of melanocytic lesions. Melanoma Research. <https://doi.org/10.1097/00008390-200404000-00008>

- [12] Zortea, M., Schopf, T. R., Thon, K., Geilhufo, M., Hindberg, K., Kirchesch, H., ... Godtlielsen, F. (2014). Performance of a dermoscopy-based computer vision system for the diagnosis of pigmented skin lesions compared with visual evaluation by experienced dermatologists. *Artificial Intelligence in Medicine*. <https://doi.org/10.1016/j.artmed.2013.11.006>
- [13] Stoecker, W. V., Gupta, K., Stanley, R. J., Moss, R. H., & Shrestha, B. (2005). Detection of asymmetric blotches (asymmetric structureless areas) in dermoscopy images of malignant melanoma using relative color. *Skin Research and Technology*. <https://doi.org/10.1111/j.1600-0846.2005.00117.x>
- [14] Madasu, V. K., & Lovell, B. C. (2009). Blotch Detection in Pigmented Skin Lesions Using Fuzzy Co-clustering and Texture Segmentation. In *2009 Digital Image Computing: Techniques and Applications*. <https://doi.org/10.1109/DICTA.2009.15>
- [15] Celebi, M. E., Kingravi, H. A., Aslandogan, Y. A., & Stoecker, W. V. (2006). Detection of blue-white veil areas in dermoscopy images using machine learning techniques, (March), 61445T. <https://doi.org/10.1117/12.655779>
- [16] Di Leo, G., Fabbrocini, G., Paolillo, A., Rescigno, O., & Sommella, P. (2009). Towards an automatic diagnosis system for skin lesions: Estimation of blue-whitish veil and regression structures. In *2009 6th International Multi-Conference on Systems, Signals and Devices, SSD 2009*. <https://doi.org/10.1109/SSD.2009.4956770>
- [17] De Vita, V., Di Leo, G. D. L., Fabbrocini, G., Liguori, C., Paolillo, A., & Sommella, P. (2012). Statistical Techniques Applied to the Automatic Diagnosis of Dermoscopic Images. *Acta Imeko*, 1(1), 7–18. Retrieved from [http://acta.imeko.org/index.php/acta-imeko/article/view/IMEKO-ACTA-01\(2012\)-01-06](http://acta.imeko.org/index.php/acta-imeko/article/view/IMEKO-ACTA-01(2012)-01-06)
- [18] Figueiredo, M. A. T., & Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://doi.org/10.1109/34.990138>
- [19] Xu, L., & Jordan, M. I. (1996). On Convergence Properties of the EM Algorithm for Gaussian Mixtures. *Neural Computation*. <https://doi.org/10.1162/neco.1996.8.1.129>
- [20] Muthén, B., & Shedden, K. (1999). Finite mixture modeling with mixture outcomes using the EM algorithm. *Biometrics*. <https://doi.org/10.1111/j.0006-341X.1999.00463.x>
- [21] Roberts, S. J., Husmeier, D., Rezek, I., & Penny, W. (1998). Bayesian approaches to Gaussian mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://doi.org/10.1109/34.730550>
- [22] Rissanen, J. (1989). *Stochastic Complexity in Statistical Inquiry*. Singapore: World Scientific
- [23] Banfield, J. D., & Raftery, A. E. (1993). Model-Based Gaussian and Non-Gaussian Clustering. *Biometrics*. <https://doi.org/10.2307/2532201>
- [24] Oliver, J. J., Baxter, R. a., & Wallace, C. S. (1996). Unsupervised Learning Using MML. *Proc. 13th Int'l Conf. Machine Learning (ICML'96)*.
- [25] Cover, T. M., & Thomas, J. A. (1991). *Elements of Information Theory*. Book. <https://doi.org/10.1177/0022219410375001>

-
- [26] Celeux, G., Chretien, S., Forbes, F., & Mkhadri, A. (2001). A component-wise EM algorithm for mixtures. *Journal of Computational and Graphical Statistics*. <https://doi.org/10.1198/106186001317243403>
- [27] García, M., Acha, B. & Gotarredona, C. (2017). Trabajo de Fin de Grado en Ingeniería de las Tecnologías de Telecomunicación Algoritmo de segmentación de lesiones pigmentadas de la piel basado en minimización de energías.
- [28] Arámbula, F.(2008) Análisis de imágenes médicas: Análisis manual de imágenes médicas:Análisis de la curva ROC (Receiver Operating Characteristic).
- [29] Pérez, A. B. (2016). Desarrollo de algoritmos de procesamiento de imagen avanzado para interpretación de imágenes médicas. Aplicación a segmentación de hígado sobre imágenes de Resonancia Magnética multisección, 2016(cc), 230.
- [30] Arámbula, F.(2008) Análisis de imágenes médicas: Análisis manual de imágenes médicas: Validación de técnicas automáticas de análisis de imágenes médicas.
- [31] Barata, C., Celebi, M. E., & Marques, J. S. (2015). Improving dermoscopy image classification using color constancy. *IEEE Journal of Biomedical and Health Informatics*. <https://doi.org/10.1109/JBHI.2014.2336473>
- [32] Finlayson, G. D., & Trezzi, E. (2004). Shades of gray and colour constancy. *Proceedings of the Twelfth Color Imaging Conference*. <https://doi.org/10.1353/hcr.0.0118>
- [33] Satheesha, T. Y., Satyanarayana, D., Prasad, M. N. G., & Dhruve, K. D. (2017). Melanoma Is Skin Deep: A 3D Reconstruction Technique for Computerized Dermoscopic Skin Lesion Classification. *IEEE Journal of Translational Engineering in Health and Medicine*. <https://doi.org/10.1109/JTEHM.2017.2648797>
- [34] Chatterjee, S., Dey, D., & Munshi, S. (2018). Development of a superpixel based local color feature extraction technique for the classification of skin lesions. In *2017 IEEE Calcutta Conference, CALCON 2017 - Proceedings (Vol. 2018-Janua, pp. 210–214)*. <https://doi.org/10.1109/CALCON.2017.8280726>

ANEXO A: CÓDIGOS

En este Anexo se presentan las diferentes funciones utilizadas para evaluar, analizar y extraer los resultados expuestos en el capítulo 4 de este trabajo. La descripción realizada de dichas funciones está escrita en inglés debido a que son las proporcionadas por [1]. A las mismas, solo se les han realizado ciertas modificaciones con el fin de cumplir las expectativas del trabajo, pero en ningún caso afectan al resultado que se puede extraer de ellas.

A.1 Código 1

```
function [color_mixture] = Color_Mixture_Pallete( X,k_min,
k_max,regularize_cov,stop_th,covoption,replication,color_id)
%Allows the estimation of a Gaussian mixture model to represent the
%dermoscopic colors, as describe in the paper
% @inproceedings{barata2014color,
%   title={Color identification in dermoscopy images using gaussian mixture
models},
%   author={Barata, C. and Figueiredo, M{. A. T. and Celebi, M. E. and
Marques, J. S.},
%   booktitle={Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE
International Conference on},
%   pages={3611--3615},
%   year={2014},
%   organization={IEEE}
% }

%The mixtures are estimated using the algorithm described in
% @article{figueiredo2002unsupervised,
%   title={Unsupervised learning of finite mixture models},
%   author={Figueiredo, M. A. T. and Jain, A. K.},
%   journal={IEEE Transactions on pattern analysis and machine intelligence},
%   volume={24},
%   number={3},
%   pages={381--396},
%   year={2002},
%   publisher={Ieee}
% }

% Inputs:
% "X" is the training data: cell structure with the names of the training
% color patches (e.g., patch_blue_gray_1.jpg)
%
% "k_max" is the initial (maximum) number of mixture components
% "k_min" is the minimum number of components to be tested
%
% "covoption" controls the covarince matrix
% covoption = 0 means free covariances
% covoption = 1 means diagonal covariances
% covoption = 2 means a common covariance for all components
% covoption = 3 means a common diagonal covariance for all components
%
```

```

% "regularize_cov," is a regularizing factor for covariance matrices; in very
small samples,
% it may be necessary to add some small quantity to the diagonal of the
covariances
%
% "stop_th" is a stopping threshold
%
% "replication" is the number of times the estimations algorithm runs. If
% replication > 1, then the best mixture is selected.
%
% "color_id" identifies the estimated color pallet:
%         'BG' - blue_gray
%         'DB' - dark brown
%         'LB' - light brown
%         'B'  - black
%         'R'  - red
%         'W'  - white

% Outputs:
% "color_mixture" is a structure that contains the following information
% "color_mixture.bestk" is the selected number of components
% "color_mixture.bestpp" is the obtained vector of mixture probabilities
% "color_mixture.bestmu" contains the estimates of the means of the
components
%         it has bestk columns by d lines
% "color_mixture.bestcov" contains the estimates of the covariances of the
components
%         it is a three-dimensional (three indexes) array
%         such that bestcov(:,:,1) is the d by d covariance of the first
%         component and bestcov(:,:,bestk) is the covariance of the last
%         component

train_data = [];
for d=1:length(X)
    I = imread(X{d});
    I = im2double(I);

    Ilab = rgb2lab(I);

    Ihsv = rgb2hsv(I);

    train_data(:,d) =
[mean2(Ilab(:,:,1));mean2(Ilab(:,:,2));mean2(Ilab(:,:,3));mean2(Ihsv(:,:,1));
mean2(Ihsv(:,:,2));mean2(Ihsv(:,:,3))];
end
%Para guardar el último elemento de cada vector de bestdlf
ultimo=[];
for iter = 1: replication
    close all
    [bestk,bestpp,bestmu,bestcov,dl,countf] =
mixtures4(train_data,k_min,k_max,regularize_cov,stop_th,covoption);

    bestkf{iter} = bestk;
    bestppf{iter} = bestpp;
    bestmuf{iter} = bestmu;
    bestcovf{iter} = bestcov;
    bestdlf{iter} = dl;
    bestcountf{iter} = countf;
    % Se toma el último valor de cada iteración--> mínimo relativo, daría
    % igual coger el valor de cualquier iteración
    vector=bestdlf{1,iter};

```

```

ultimo(1,iter)=vector(1,end);

end

% Se toma el menor valor de dl en módulo
[minValue, indexOfMin] = min(ultimo(1,:));
model=indexOfMin;

color_mixture.bestk = bestkf{model};
color_mixture.bestpp = bestppf{model};
color_mixture.bestmu = bestmuf{model};
color_mixture.bestcov = bestcovf{model};

save([color_id '_model'], 'color_mixture');
```

A.2 Código 2

```

function [color_mask, detected_colors] = Detect_Colors( img, step, seg)
%Allows the detection of dermoscopic colors using Gaussian mixture
%models as described in
% @inproceedings{barata2014color,
%   title={Color identification in dermoscopy images using gaussian mixture
models},
%   author={Barata, C. and Figueiredo, M{. A. T. and Celebi, M. E. and
Marques, J. S.},
%   booktitle={Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE
International Conference on},
%   pages={3611--3615},
%   year={2014},
%   organization={IEEE}
% }

% Inputs:
% "img" is the test imagen in uint8 format
%
% "step" is the size of the patches to be labeled. If step = 1, the labeling
is pixel
% wise
%
% "seg" is the binary segmentation mask of the lesion

% Outputs:
% "color_mask" is a mask of the pixel/patches labels
% "detected_colors" identifies the observable colors (1-present/0-absent).
% The color sorting of the vector is the following: BG, DB, LB, B, W, R

I = im2double(img);

NBlock=step+2;

score_bg = [0];
score_db = [0];
score_lb = [0];
```

```

score_b = [0];
score_w = [0];
score_r = [0];

seg = imerode(seg,strel('disk',3));

w = I<=0.03928;
I(w) = I(w)/12.92;
I(~w) = ((I(~w)+.055)/1.055).^2.2;

s_R = power(sum(sum(I(:, :, 1).^3)/(size(I,1)*size(I,2))),1/3);
s_G = power(sum(sum(I(:, :, 2).^3)/(size(I,1)*size(I,2))),1/3);
s_B = power(sum(sum(I(:, :, 3).^3)/(size(I,1)*size(I,2))),1/3);

som = sqrt(s_R.^2+s_G.^2+s_B.^2);

s_R = s_R/som;
s_G = s_G/som;
s_B = s_B/som;

I(:, :, 1) = I(:, :, 1)./(s_R*sqrt(3));
I(:, :, 2) = I(:, :, 2)./(s_G*sqrt(3));
I(:, :, 3) = I(:, :, 3)./(s_B*sqrt(3));

Ilab = rgb2lab(I);

Ihsv = rgb2hsv(I);

M2 = zeros(size(I,1),size(I,2));

NI=size(I,1); MI=size(I,2); % image size

t=1;

for p=ceil(NBlock/2):step:NI-ceil((NBlock-1)/2), % block features
    for q=ceil(NBlock/2):step:MI-ceil((NBlock-1)/2),

        l = floor(NBlock/2);
        vi = p-l:p+l; % es una L
        vj = q-l:q+l; % es una L

        if sum(sum(seg(vi,vj))) == (NBlock^2)

            I_aux = Ilab(vi,vj,1:3);

            L1 = mean2(I_aux(:, :, 1));

            L2 = mean2(I_aux(:, :, 2));

            L3 = mean2(I_aux(:, :, 3));
            %
            I_aux = Ihsv(vi,vj,1:3);

            L4 = mean2(I_aux(:, :, 1));

            L5 = mean2(I_aux(:, :, 2));

            L6 = mean2(I_aux(:, :, 3));

```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%BLUE GRAY%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load BG_model
for k = 1:color_mixture.bestk

yg=multinorm([L1;L2;L3;L4;L5;L6],color_mixture.bestmu(:,k),color_mixture.best
cov(:, :, k));
    score_bg(t) = score_bg(t)+color_mixture.bestpp(k)*yg;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%DARK BROWN%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load DB_model

for k = 1:color_mixture.bestk

yg=multinorm([L1;L2;L3;L4;L5;L6],color_mixture.bestmu(:,k),color_mixture.best
cov(:, :, k));
    score_db(t) = score_db(t)+color_mixture.bestpp(k)*yg;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%LIGHT BROWN%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load LB_model

for k = 1:color_mixture.bestk

yg=multinorm([L1;L2;L3;L4;L5;L6],color_mixture.bestmu(:,k),color_mixture.best
cov(:, :, k));
    score_lb(t) = score_lb(t)+color_mixture.bestpp(k)*yg;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%BLACK%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load B_model

for k = 1:color_mixture.bestk

yg=multinorm([L1;L2;L3;L4;L5;L6],color_mixture.bestmu(:,k),color_mixture.best
cov(:, :, k));
    score_b(t) = score_b(t)+color_mixture.bestpp(k)*yg;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%WHITE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load W_model

for k = 1:color_mixture.bestk

yg=multinorm([L1;L2;L3;L4;L5;L6],color_mixture.bestmu(:,k),color_mixture.best
cov(:, :, k));
    score_w(t) = score_w(t)+color_mixture.bestpp(k)*yg;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%RED%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load R_model

for k = 1:color_mixture.bestk

```

```

yg=multinorm([L1;L2;L3;L4;L5;L6],color_mixture.bestmu(:,k),color_mixture.best
cov(:, :,k));
    score_r(t) = score_r(t)+color_mixture.bestpp(k)*yg;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIND COLOR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
scores = [score_bg(t), score_db(t),score_lb(t), score_b(t),
score_w(t), score_r(t)];
scores = scores./(sum(scores)+realmin);
[v,c1] = max(scores);
ths = .8;
others = find(scores > ths*v);
if length(others)==1
    M2(vi,vj) = c1;
elseif length(others)>2 || v < .35
    M2(vi,vj) = 22;
else %length(others)==2
    c2 = setdiff(others,c1);
    c = sort([c1 c2]);
    Mlabels = zeros(6);
    Mlabels(find(tril(ones(6),-1))) = (7:21)';
    M2(vi,vj) = Mlabels(c(2),c(1));
end

t=t+1;
%           score_skin = [score_skin 0];
score_bg = [score_bg 0];
score_db = [score_db 0];
score_lb = [score_lb 0];
score_b = [score_b 0];
score_w = [score_w 0];
score_r = [score_r 0];

end
end
end

A = sum(sum(seg));

BG = sum(sum(M2 == 1))/A;

DB = sum(sum(M2 == 2))/A;

LB = sum(sum(M2 == 3))/A;

B = sum(sum(M2 == 4))/A;

W = sum(sum(M2==5))/A;

R = sum(sum(M2==6))/A;

%
[ color_mask, detected_colors] = Quantify_Colors( BG, DB, LB, B, W,R, M2 );
end

```

A.3 Código 3

```
function [ color_mask, detected_colors ] = Quantify_Colors (BG, DB, LB, B,
W,R, M)
%Allows the quantification of dermoscopic colors using Gaussian mixture
%models as described in
% @inproceedings{barata2014color,
%   title={Color identification in dermoscopy images using gaussian mixture
models},
%   author={Barata, C. and Figueiredo, M{. A. T. and Celebi, M. E. and
Marques, J. S.},
%   booktitle={Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE
International Conference on},
%   pages={3611--3615},
%   year={2014},
%   organization={IEEE}
% }

% Inputs:
% "BG, DB, LB, B, W, R" are the masks of each color
%
% "M" is the global pixel/patch color labeling mask
%

% Outputs:
% "color_mask" is the new and corrected mask of the pixel/patches labels
% "detected_colors" identifies the observable colors (1-present/0-absent).
% The color sorting of the vector is the following: BG, DB, LB, B, W, R

color_mask = M;

detected_colors = [1 1 1 1 1 1];

if BG < 0.04 %Threshold can be tuned
    color_mask(M==1) = 0;
    detected_colors(1) = 0;
end

if DB < 0.05 %Threshold can be tuned
    color_mask(M==2) = 0;
    detected_colors(2) = 0;
end

if LB < 0.15 %Threshold can be tuned
    color_mask (M==3) = 0;
    detected_colors(3) = 0;
end

if B < 0.02 %Threshold can be tuned
    color_mask(M==4) = 0;
    detected_colors(4) = 0;
end

if W < 0.02 %Threshold can be tuned
    color_mask(M==5) = 0;
    detected_colors(5) = 0;
end

if R < 0.03 %Threshold can be tuned
    color_mask(M==6) = 0;
```

```

        detected_colors(6) = 0;
end

end

```

A.4 Código 4

```

function [bestk,bestpp,bestmu,bestcov,dl,countf] =
mixtures4(y,kmin,kmax,regularize,th,covoption)
%
%
% Usage syntax:
% [bestk,bestpp,bestmu,bestcov,dl,countf] =
mixtures3(y,kmin,kmax,regularize,th,covoption)
%
% Inputs:
%
% "y" is the data; for n observations in d dimensions, y must have
% d lines and n columns.
%
% "kmax" is the initial (maximum) number of mixture components
% "kmin" is the minimum number of components to be tested
%
% "covoption" controls the covarince matrix
% covoption = 0 means free covariances
% covoption = 1 means diagonal covariances
% covoption = 2 means a common covariance for all components
% covoption = 3 means a common diagonal covarince for all components
%
% "regularize" is a regularizing factor for covariance matrices; in very
small samples,
% it may be necessary to add some small quantity to the diagonal of the
covariances
%
% "th" is a stopping threshold
%
% Outputs:
%
% "bestk" is the selected number of components
% "bestpp" is the obtained vector of mixture probabilities
% "bestmu" contains the estimates of the means of the components
%         it has bestk columns by d lines
% "bestcov" contains the estimates of the covariances of the components
%         it is a three-dimensional (three indexes) array
%         such that bestcov(:, :, 1) is the d by d covariance of the first
%         component and bestcov(:, :, bestk) is the covariance of the last
%         component
%
% "dl" contains the successive values of the cost function
% "countf" is the total number of iterations performed
%
% Written by:
%   Mario Figueiredo
%   Instituto Superior Tecnico
%   1049-001 Lisboa
%   Portugal
%
%   Email: mtf@lx.it.pt

```

```

%
% 2000, 2001, 2002
%
% -----
% Copyright (2002): Mario A. T. Figueiredo and Anil K. Jain
%
% This software is distributed under the terms
% of the GNU General Public License 2.0.
%
% Permission to use, copy, and distribute this software for
% any purpose without fee is hereby granted, provided that this entire
% notice is included in all copies of any software which is or includes
% a copy or modification of this software and in all copies of the
% supporting documentation for such software.
% This software is being provided "as is", without any express or
% implied warranty. In particular, the authors do not make any
% representation or warranty of any kind concerning the merchantability
% of this software or its fitness for any particular purpose."
% -----
%
verb=1; % verbose mode; change to zero for silent mode
bins = 40; % number of bins for the univariate data histograms for
visualization
dl = []; % vector to store the consecutive values of the cost function
[dimensions, npoints] = size(y);

switch covoption
    case 0
        npars = (dimensions + dimensions*(dimensions+1)/2);
        %this is for free covariance matrices
    case 1
        npars = 2*dimensions;
        %this is for diagonal covariance matrices
    case 2
        npars = dimensions;
        %this is for a common covariance matrix
        %independently of its structure)
    case 3
        npars = dimensions;
    otherwise
        %the default is to assume free covariances
        npars = (dimensions + dimensions*(dimensions+1)/2);
end
nparsover2 = npars / 2;

% we choose which axes to use in the plot,
% in case of higher dimensional data (>2)
% Change this to have other axes being shown
axis1 = 1;
axis2 = 2;

% kmax is the initial number of mixture components
k = kmax;

% indic will contain the assignments of each data point to
% the mixture components, as result of the E-step
indic = zeros(k, npoints);

% Initialization: we will initialize the means of the k components
% with k randomly chosen data points. Randperm(n) is a MATLAB function
% that generates random permutations of the integers from 1 to n.

```

```

randindex = randperm(npoints);
randindex = randindex(1:k);
estmu = y(:,randindex);

% the initial estimates of the mixing probabilities are set to 1/k
estpp = (1/k)*ones(1,k);

% here we compute the global covariance of the data
globcov = cov(y');

for i=1:k
    % the covariances are initialized to diagonal matrices proportional
    % to 1/10 of the mean variance along all the axes.
    % Of course, this can be changed
    estcov(:, :, i) = diag(ones(1, dims) * max(diag(globcov/10)));
end

% we plot the data and the initial estimates
if (dims == 1)
    [hh, xx] = hist(y, bins);
    barplot = bar(xx, hh/npoints/(xx(2)-xx(1)));
    set(barplot, 'EdgeColor', [1 1 1]);
    set(barplot, 'FaceColor', [0.75 0.75 0.75]);
    set(gca, 'FontName', 'Times'); set(gca, 'FontSize', 14);
    drawnow
    hold on
    miny = min(y);
    maxy = max(y);
    plotgrid = miny:(maxy-miny)/200:maxy;
    mix = zeros(size(plotgrid));
    for comp=1:k
        mix = mix + estpp(comp)*uninorm(plotgrid, estmu(comp), estcov(comp));
    end
    plot(plotgrid, estpp(comp)*uninorm(plotgrid, estmu(comp), estcov(comp)), 'k')
    plot(plotgrid, mix, 'Color', 'red', 'LineWidth', 3);
    text(plotgrid(5), 0.9*max(mix), sprintf('k=%d', k), ...
        'FontName', 'Times', 'FontSize', 18, 'FontWeight', 'bold');
    drawnow
    hold off
else
    plot(y(axis1, :), y(axis2, :), '.', 'Color', [0.5 0.5 0.5]);
    hold on
    axis equal
    set(gca, 'FontName', 'Times', 'FontSize', 22);
    placex = get(gca, 'Xlim'); placey = get(gca, 'Ylim');
    text(placex(1)+0.2, placey(2)-0.2, sprintf('k=%d', k), ...
        'FontName', 'Times', 'FontSize', 22);
    for comp=1:k
        elipsnorm(estmu([axis1, axis2], comp), ...
            estcov([axis1, axis2], [axis1, axis2], comp), 2)
    end
    drawnow
    hold off
end

% having the initial means, covariances, and probabilities, we can
% initialize the indicator functions following the standard EM equation
% Notice that these are unnormalized values
for i=1:k

```

```

    semi_indic(i,:) = multinorm(y,estmu(:,i),estcov(:,:,i));
    indic(i,:) = semi_indic(i,:)*estpp(i);
end

% we can use the indic variables (unnormalized) to compute the
% loglikelihood and store it for later plotting its evolution
% we also compute and store the number of components
countf = 1;
loglike(countf) = sum(log(sum(realmin+indic)));
dlength = -loglike(countf) + (nparsover2*sum(log(estpp))) + ...
    (nparsover2 + 0.5)*k*log(npoints);
dl(countf) = dlength;
kappas(countf) = k;

% the transitions vectors will store the iteration
% number at which components are killed.
% transitions1 stores the iterations at which components are
% killed by the M-step, while transitions2 stores the iterations
% at which we force components to zero.
transitions1 = [];
transitions2 = [];

% minimum description length seen so far, and corresponding
% parameter estimates
mindl = dl(countf);
bestpp = estpp;
bestmu = estmu;
bestcov = estcov;
bestk = k;

k_cont = 1;    % auxiliary variable for the outer loop
while(k_cont) % the outer loop will take us down from kmax to kmin
components
    cont=1;    % auxiliary variable of the inner loop
    while(cont) % this inner loop is the component-wise EM algorithm with
the
        % modified M-step that can kill components
        if verb==1
            % in verbose mode, we keep displaying the minimum of the
            % mixing probability estimates to see how close we are
            % to killing one component
            disp(sprintf('k = %2d,  minestpp = %0.5g', k, min(estpp)));
        end

        % we begin at component 1
        comp = 1;
        % ...and can only go to the last component, k.
        % Since k may change during the process, we can not use a for loop
        while comp <= k
            % we start with the M step
            % first, we compute a normalized indicator function
            clear indic
            for i=1:k
                indic(i,:) = semi_indic(i,:)*estpp(i);
            end
            normindic = indic./(realmin+kron(ones(k,1),sum(indic,1)));

            % now we perform the standard M-step for mean and covariance
            normalize = 1/sum(normindic(comp,:));
            aux = kron(normindic(comp,:),ones(dimens,1)).*y;

```

```

    estmu(:,comp)= normalize*sum(aux,2);
    if (covoption == 0)|| (covoption == 2)
        estcov(:, :, comp) = normalize*(aux*y') -
estmu(:,comp)*estmu(:,comp)' ...
        + regularize*eye(dimens);
    else
        estcov(:, :, comp) = normalize*diag(sum(aux.*y,2)) -
diag(estmu(:,comp).^2) ;
    end
    if covoption == 2
        comcov = zeros(dimens,dimens);
        for comp2 = 1:k
            comcov = comcov + estpp(comp2)*estcov(:, :, comp2);
        end
        for comp2 = 1:k
            estcov(:, :, comp2) = comcov;
        end
    end
    if covoption == 3
        comcov = zeros(dimens,dimens);
        for comp2 = 1:k
            comcov = comcov +
estpp(comp2)*diag(diag(estcov(:, :, comp2)));
        end
        for comp2 = 1:k
            estcov(:, :, comp2) = comcov;
        end
    end
end

% this is the special part of the M step that is able to
% kill components
estpp(comp) = max(sum(normindic(comp, :))-nparsover2,0)/npoints;
estpp = estpp/sum(estpp);

% this is an auxiliary variable that will be used the
% signal the killing of the current component being updated
killed = 0;

% we now have to do some book-keeping if the current component
was killed
% that is, we have to rearrange the vectors and matrices that
store the
% parameter estimates
if estpp(comp)==0
    killed = 1;
    % we also register that at the current iteration a component
was killed
    transitions1 = [transitions1 countf];
    if comp==1
        estmu = estmu(:,2:k);
        estcov = estcov(:, :, 2:k);
        estpp = estpp(2:k);
        semi_indic = semi_indic(2:k, :);
    else
        if comp==k
            estmu = estmu(:,1:k-1);
            estcov = estcov(:, :, 1:k-1);
            estpp = estpp(1:k-1);
            semi_indic = semi_indic(1:k-1, :);
        else

```



```

        estmu = [estmu(:,1:comp-1) estmu(:,comp+1:k)];
        newcov = zeros(dimens,dimens,k-1);
        for kk=1:comp-1
            newcov(:, :, kk) = estcov(:, :, kk);
        end
        for kk=comp+1:k
            newcov(:, :, kk-1) = estcov(:, :, kk);
        end
        estcov = newcov;
        estpp = [estpp(1:comp-1) estpp(comp+1:k)];
        semi_indic = semi_indic([1:comp-1, comp+1:k], :);
    end
end

    % since we've just killed a component, k must decrease
    k=k-1;
end

    if (covoption == 2) || (covoption == 3)
        for kk = 1:k
            semi_indic(kk, :) =
multinorm(y, estmu(:, kk), estcov(:, :, kk));
        end
    end

    if killed==0
        % if the component was not killed, we update the
corresponding
        % indicator variables...
        semi_indic(comp, :) =
multinorm(y, estmu(:, comp), estcov(:, :, comp));
        % ...and go on to the next component
        comp = comp + 1;
    end
    % if killed==1, it means the in the position "comp", we now
    % have a component that was not yet visited in this sweep, and
    % so all we have to do is go back to the M setp without
    % increasing "comp"

end % this is the end of the innermost "while comp <= k" loop
% which cycles through the components

% increment the iterations counter
countf = countf + 1;

clear indic
clear semi_indic
for i=1:k
    semi_indic(i, :) = multinorm(y, estmu(:, i), estcov(:, :, i));
    indic(i, :) = semi_indic(i, :)*estpp(i);
end

if k~=1
    % if the number of surviving components is not just one, we
compute
    % the loglikelihood from the unnormalized assignment variables
    loglike(countf) = sum(log(realmin+sum(indic)));
else
    % if it is just one component, it is even simpler
    loglike(countf) = sum(log(realmin+indic));
end

```

```

        % compute and store the description length and the current number of
components
        dlength = -loglike(countf) + (nparsover2*sum(log(estpp))) + ...
            (nparsover2 + 0.5)*k*log(npoints);
        dl(countf) = dlength;
        kappas(countf) = k;

        % compute the change in loglikelihood to check if we should stop
        deltlke = loglike(countf) - loglike(countf-1);
        if (verb~=0)
            disp(sprintf('deltaloglike/th = %0.7g',
abs(deltlike/loglike(countf-1))/th));
        end
        if (abs(deltlike/loglike(countf-1)) < th)
            % if the relative change in loglikelihood is below the threshold,
we stop CEM2
            cont=0;
        end

    end% this end is of the inner loop: "while(cont)"

figure
if dimens == 1
    [hh,xx] = hist(y,bins);
    barplot = bar(xx,hh/npoints/(xx(2)-xx(1)));
    set(barplot,'EdgeColor',[1 1 1]);
    set(barplot,'FaceColor',[0.75 0.75 0.75]);
    set(gca,'FontName','Times'); set(gca,'FontSize',14);
    drawnow
    hold on
    miny = min(y);
    maxy = max(y);
    plotgrid = miny:(maxy-miny)/200:maxy;
    mix = zeros(size(plotgrid));
    for comp=1:k
        mix = mix +
estpp(comp)*uninorm(plotgrid,estmu(comp),estcov(comp));

plot(plotgrid,estpp(comp)*uninorm(plotgrid,estmu(comp),estcov(comp)),'k')
    end
    plot(plotgrid,mix,'Color','red','LineWidth',3);
    text(plotgrid(5),0.9*max(mix),sprintf('k=%d',k),...
        'FontName','Times','FontSize',18,'FontWeight','bold');
    drawnow
    hold off
else
    plot(y(axis1,:),y(axis2,:),'.','Color',[0.5 0.5 0.5]);
    placex = get(gca,'Xlim'); placey = get(gca,'Ylim');
    text(placex(1)+0.2,placey(2)-0.2,sprintf('k=%d',k),...
        'FontName','Times','FontSize',22);
    hold on
    axis equal
    set(gca,'FontName','Times','FontSize',22);
    for comp=1:k

elipsnorm(estmu([axis1,axis2],comp),estcov([axis1,axis2],[axis1,axis2],comp),
2)

    end
    drawnow
    hold off

```

```

end

% now check if the latest description length is the best;
% if it is, we store its value and the corresponding estimates
if dl(countf) < mindl
    bestpp = estpp;
    bestmu = estmu;
    bestcov = estcov;
    bestk = k;
    mindl = dl(countf);
end

% at this point, we may try smaller mixtures by killing the
% component with the smallest mixing probability and then restarting
CEM2,
% as long as k is not yet at kmin
if k>kmin
    [minp ,indminp] = min(estpp);
    % what follows is the book-keeping associated with removing one
component
    if indminp==1
        estmu = estmu(:,2:k);
        estcov = estcov(:,:,2:k);
        estpp = estpp(2:k);
    else
        if indminp==k
            estmu = estmu(:,1:k-1);
            estcov = estcov(:,:,1:k-1);
            estpp = estpp(1:k-1);
        else
            estmu = [estmu(:,1:indminp-1) estmu(:,indminp+1:k)];
            newcov = zeros(dimens,dimens,k-1);
            for kk=1:indminp-1
                newcov(:,:,kk) = estcov(:,:,kk);
            end
            for kk=indminp+1:k
                newcov(:,:,kk-1) = estcov(:,:,kk);
            end
            estcov = newcov;
            estpp = [estpp(1:indminp-1) estpp(indminp+1:k)];
        end
    end
    k=k-1;

    % we renormalize the mixing probabilities after killing the component
    estpp = estpp/sum(estpp);

    % and register the fact that we have forced one component to zero
    transitions2 = [transitions2 countf];

    %increment the iterations counter
    countf = countf+1;

    % ...and compute the loglikelihood function and the description
length
    clear indic
    clear semi_indic
    for i=1:k
        semi_indic(i,:) = multinorm(y,estmu(:,i),estcov(:,:,i));
    end
end

```

```

        indic(i,:) = semi_indic(i,:)*estpp(i);
    end
    if k~=1
        loglike(countf) = sum(log(realmin+sum(indic)));
    else
        loglike(countf) = sum(log(realmin+indic));
    end
    dl(countf) = -loglike(countf) + (nparsover2*sum(log(estpp))) + ...
        (nparsover2 + 0.5)*k*log(npoints);

    kappas(countf) = k;

    else %this else corresponds to "if k > kmin"
        %of course, if k is not larger than kmin, we must stop
        k_cont = 0;
    end

end % this is the end of the outer loop "while(k_cont)"

lastpp = estpp;
lastmu = estmu;
lastcov = estcov;

% finally, we plot the results
figure
if dims == 1
    [hh,xx] = hist(y,bins);
    barplot = bar(xx,hh/npoints/(xx(2)-xx(1)));
    set(barplot,'EdgeColor',[1 1 1]);
    set(barplot,'FaceColor',[0.75 0.75 0.75]);
    set(gca,'FontName','Times'); set(gca,'FontSize',14);
    drawnow
    hold on
    miny = min(y);
    maxy = max(y);
    plotgrid = miny:(maxy-miny)/200:maxy;
    mix = zeros(size(plotgrid));
    for comp=1:bestk
        mix = mix +
bestpp(comp)*uninorm(plotgrid,bestmu(comp),bestcov(comp));

plot(plotgrid,bestpp(comp)*uninorm(plotgrid,bestmu(comp),bestcov(comp)),'k')
    end
    plot(plotgrid,mix,'Color','red','LineWidth',3);
    text(plotgrid(5),0.9*max(mix),sprintf('k=%d',bestk),...
        'FontName','Times','FontSize',18,'FontWeight','bold');
    drawnow
    hold off
else
    plot(y(axis1,:),y(axis2:,:),'.','Color',[0.5 0.5 0.5]);
    hold on
    axis equal
    set(gca,'FontName','Times','FontSize',22);
    placex = get(gca,'Xlim'); placey = get(gca,'Ylim');
    text(placex(1)+1,placey(2)-1,sprintf('k=%d',length(bestpp)),...
        'FontName','Times','FontSize',22);
    for comp=1:length(bestpp)

elipsnorm(bestmu([axis1,axis2],comp),bestcov([axis1,axis2],[axis1,axis2],comp
),2)

```

```

        end
        drawnow
        hold off
    end
    thefig=gcf;

    % now, we plot the evolution of the description length
    % and signal where components were killed
    figure
    plot(dl, 'LineWidth', 2)
    title('Description Length')
    ax = axis;
    for i=1:length(transitions1)
        line([transitions1(i) transitions1(i)], [ax(3) ax(4)])
    end
    for i=1:length(transitions2)
        line([transitions2(i) transitions2(i)], [ax(3) ax(4)], 'LineStyle', ':')
    end
    figure(thefig)

```

A.5 Código 5

```

function y = elipsnorm(m,cov,level,dashed)
%
%
% draws one contour plot of a bivariate
% gaussian density with mean "m" and covariance
% matrix "cov".
% The level is controled by "level".
% If "dashed==1", the line is dashed.
%
% -----
% Copyright (2002): Mario A. T. Figueiredo and Anil K. Jain
%
% This software is distributed under the terms
% of the GNU General Public License 2.0.
%
% Permission to use, copy, and distribute this software for
% any purpose without fee is hereby granted, provided that this entire
% notice is included in all copies of any software which is or includes
% a copy or modification of this software and in all copies of the
% supporting documentation for such software.
% This software is being provided "as is", without any express or
% implied warranty. In particular, the authors do not make any
% representation or warranty of any kind concerning the merchantability
% of this software or its fitness for any particular purpose."
% -----

if nargin<4
    dashed=0;
end
[uu,ei,vv]=svd(cov);
a = sqrt(ei(1,1)*level*level);
b = sqrt(ei(2,2)*level*level);
theta = [0:0.01:2*pi];
xx = a*cos(theta);
yy = b*sin(theta);
cord = [xx' yy']';

```

```

cord = uu*cord;
if dashed==1
    plot(cord(1,:)+m(1),cord(2,:)+m(2),'--k','LineWidth',2)
else
    plot(cord(1,:)+m(1),cord(2,:)+m(2),'k','LineWidth',2)
end

```

A.6 Código 6

```

function y = multinorm(x,m,covar)
% Evaluates a multidimensional Gaussian
% of mean m and covariance matrix covar
% at the array of points x
%
% -----
% Copyright (2002): Mario A. T. Figueiredo and Anil K. Jain
%
% This software is distributed under the terms
% of the GNU General Public License 2.0.
%
% Permission to use, copy, and distribute this software for
% any purpose without fee is hereby granted, provided that this entire
% notice is included in all copies of any software which is or includes
% a copy or modification of this software and in all copies of the
% supporting documentation for such software.
% This software is being provided "as is", without any express or
% implied warranty. In particular, the authors do not make any
% representation or warranty of any kind concerning the merchantability
% of this software or its fitness for any particular purpose."
% -----
%
[dim, npoints] = size(x);
dd = det(covar+ realmin*eye(dim));
in = inv(covar+ realmin*eye(dim));
ff = ((2*pi)^(-dim/2))*((dd)^(-0.5));
%quadform = zeros(1,npoints);

centered = (x-m*ones(1,npoints));
if dim ~= 1
    y = ff * exp(-0.5*sum(centered.*(in*centered)));
else
    y = ff * exp(-0.5*in*centered.^2 );
end

```

A.7 Código 7

```

function y = uninorm(x,m,var)
% Evaluates a multidimensional Gaussian
% of mean m and variance var
% at the array of points x
%
% -----
% Copyright (2002): Mario A. T. Figueiredo and Anil K. Jain

```

```
%
% This software is distributed under the terms
% of the GNU General Public License 2.0.
%
% Permission to use, copy, and distribute this software for
% any purpose without fee is hereby granted, provided that this entire
% notice is included in all copies of any software which is or includes
% a copy or modification of this software and in all copies of the
% supporting documentation for such software.
% This software is being provided "as is", without any express or
% implied warranty. In particular, the authors do not make any
% representation or warranty of any kind concerning the merchantability
% of this software or its fitness for any particular purpose."
% -----
%
ff = ((2*pi*(var+realmin))^(1/2));
y = ff * exp((-1/(2*var))*(x-m).^2);
```

A.8 Código 8

```
function y = genmix(npoints,mean,var,prob)
%
% y = genmix(npoints,mean,var,prob)
%
% Generates 'npoints' samples from a mixture
% of Gaussian densities.
% The dimension of the space is the number of
% lines of 'mean', while the number of components, 'comp',
% of the mixture is its number of columns. That is
% each column of 'mean' is the mean of one component
% of the mixture.
% Parameter 'var' must be a 3-dimensional array such that
% var(:,:,i) is the covariance matrix of the i-th
% component of the mixture. Of course, size(var(:,:,i))
% must be [dim,dim].
% 'probs', contains the mixing probabilities of the
% first comp-1 components. The comp-th one is, of course,
% 1-sum(comp)
%
% -----
% Copyright (2002): Mario A. T. Figueiredo and Anil K. Jain
%
% This software is distributed under the terms
% of the GNU General Public License 2.0.
%
% Permission to use, copy, and distribute this software for
% any purpose without fee is hereby granted, provided that this entire
% notice is included in all copies of any software which is or includes
% a copy or modification of this software and in all copies of the
% supporting documentation for such software.
% This software is being provided "as is", without any express or
% implied warranty. In particular, the authors do not make any
% representation or warranty of any kind concerning the merchantability
% of this software or its fitness for any particular purpose."
% -----
%
clear y;
getpars = size(mean);
```

```

dim = getpars(1);
modes = getpars(2);
if modes~=1
    if ( max(size(prob))~=(modes-1) | ...
        min(size(prob))~=1 | ...
        max(prob)>1 | ...
        min(prob)<0 )
        disp('Invalid vector of mixing probabilities')
        return
    end
    lastp = 1-sum(prob);
    prob = [reshape(prob,1,modes-1) lastp];
else
    prob = [1];
end
dat = [];
for i=1:modes
    if dim~=1
        filter = sqrtm(var(:, :, i));
    else
        filter = sqrt(var(i));
    end
    len = round(npoints*prob(i));
    noi = randn(dim, len);
    dat = [dat , kron(mean(:, i), ones(1, len))+filter*noi];
end
y = dat;

```

A.9 Código 9: Funciones creadas para el entrenamiento del algoritmo

```

function AZUL(covoption, regularize_cov, stop_th)
% Función para el entrenamiento del color Azul-Gris a partir de 24 parches
% obtenidos de la base de datos de imagen empleada. Finalmente, con la
% llamada de Color_Mixture_Pallete, se obtiene el modelo de mezcla
% Gaussiana para dicho color.

H={ 'blue1.bmp', 'blue2.bmp', 'blue3.bmp', 'blue4.bmp', 'blue5.bmp', 'blue6.bmp', ...
    .
    'blue7.bmp', 'blue8.bmp', 'blue9.bmp', 'blue10.bmp', 'blue11.bmp', 'blue12.bmp', ...
    .
    'blue13.bmp', 'blue14.bmp', 'blue15.bmp', 'blue16.bmp', 'blue17.bmp', 'blue18.bmp'
    , ...
    'blue19.bmp', 'blue20.bmp', 'blue21.bmp', 'blue22.bmp', 'blue23.bmp', 'blue24.bmp'
    };

% Configuración de parámetros
k_min=1;
k_max=6;
color_id='BG';
replication=3;
[color_mixture] = Color_Mixture_Pallete(H, k_min,
k_max, regularize_cov, stop_th, covoption, replication, color_id);
close all;
end

```



```

function DBROWN(covoption, regularize_cov, stop_th)
% Función para el entrenamiento del color Marrón Oscuro a partir de 24
parches
% obtenidos de la base de datos de imagen empleada. Finalmente, con la
% llamada de Color_Mixture_Pallete, se obtiene el modelo de mezcla
% Gaussiana para dicho color.

H={'dark1.bmp', 'dark2.bmp', 'dark3.bmp', 'dark4.bmp', 'dark5.bmp', 'dark6.bmp', ..
.
'dark7.bmp', 'dark8.bmp', 'dark9.bmp', 'dark10.bmp', 'dark11.bmp', 'dark12.bmp', ..
.
'dark13.bmp', 'dark14.bmp', 'dark15.bmp', 'dark16.bmp', 'dark17.bmp', 'dark18.bmp'
, ...
'dark19.bmp', 'dark20.bmp', 'dark21.bmp', 'dark22.bmp', 'dark23.bmp', 'dark24.bmp'
};

% Configuración de parámetros
k_min=1;
k_max=6;
color_id='DB';
replication=3;
[color_mixture] = Color_Mixture_Pallete(H,k_min,
k_max,regularize_cov,stop_th,covoption,replication,color_id);
close all;
end

function LBROWN(covoption, regularize_cov, stop_th)
% Función para el entrenamiento del color Marrón Claro a partir de 24 parches
% obtenidos de la base de datos de imagen empleada. Finalmente, con la
% llamada de Color_Mixture_Pallete, se obtiene el modelo de mezcla
% Gaussiana para dicho color.

H={'light1.bmp', 'light2.bmp', 'light3.bmp', 'light4.bmp', 'light5.bmp', 'light6.b
mp', ...
'light7.bmp', 'light8.bmp', 'light9.bmp', 'light10.bmp', 'light11.bmp', 'light12.b
mp', ...
'light13.bmp', 'light14.bmp', 'light15.bmp', 'light16.bmp', 'light17.bmp', 'light1
8.bmp', ...
'light19.bmp', 'light20.bmp', 'light21.bmp', 'light22.bmp', 'light23.bmp', 'light2
4.bmp'};

% Configuración de parámetros
k_min=1;
k_max=6;
color_id='LB';
replication=3;
[color_mixture] = Color_Mixture_Pallete(H,k_min,
k_max,regularize_cov,stop_th,covoption,replication,color_id);
close all;
end

```

```

function NEGRO(covoption, regularize_cov, stop_th)
% Función para el entrenamiento del color Negro a partir de 24 parches
% obtenidos de la base de datos de imagen empleada. Finalmente, con la
% llamada de Color_Mixture_Palette, se obtiene el modelo de mezcla
% Gaussiana para dicho color.

H={'negro1.bmp','negro2.bmp','negro3.bmp','negro4.bmp','negro5.bmp','negro6.b
mp',...

'negro7.bmp','negro8.bmp','negro9.bmp','negro10.bmp','negro11.bmp','negro12.b
mp',...

'negro13.bmp','negro14.bmp','negro15.bmp','negro16.bmp','negro17.bmp','negro1
8.bmp',...

'negro19.bmp','negro20.bmp','negro21.bmp','negro22.bmp','negro23.bmp','negro2
4.bmp'};

% Configuración de parámetros
k_min=1;
k_max=6;
color_id='B';
replication=3;
[color_mixture] = Color_Mixture_Palette(H,k_min,
k_max,regularize_cov,stop_th,covoption,replication,color_id);
close all;
end

```

```

function RED(covoption, regularize_cov, stop_th)
% Función para el entrenamiento del color Rojo a partir de 24 parches
% obtenidos de la base de datos de imagen empleada. Finalmente, con la
% llamada de Color_Mixture_Palette, se obtiene el modelo de mezcla
% Gaussiana para dicho color.

H={'red1.bmp','red2.bmp','red3.bmp','red4.bmp','red5.bmp','red6.bmp',...
'red7.bmp','red8.bmp','red9.bmp','red10.bmp','red11.bmp','red12.bmp',...

'red13.bmp','red14.bmp','red15.bmp','red16.bmp','red17.bmp','red18.bmp',...
'red19.bmp','red20.bmp','red21.bmp','red22.bmp','red23.bmp','red24.bmp'};

% Configuración de parámetros
k_min=1;
k_max=6;
color_id='R';
replication=3;
[color_mixture] = Color_Mixture_Palette(H,k_min,
k_max,regularize_cov,stop_th,covoption,replication,color_id);
close all;
end

```

```

function BLANCO(covoption, regularize_cov, stop_th)
% Función para el entrenamiento del color Blanco a partir de 24 parches
% obtenidos de la base de datos de imagen empleada. Finalmente, con la
% llamada de Color_Mixture_Palette, se obtiene el modelo de mezcla
% Gaussiana para dicho color.

H={'blanco1.bmp','blanco2.bmp','blanco3.bmp','blanco4.bmp','blanco5.bmp','bla
nco6.bmp',...

```

```

'blanco7.bmp','blanco8.bmp','blanco9.bmp','blanco10.bmp','blanco11.bmp','blanco12.bmp',...

'blanco13.bmp','blanco14.bmp','blanco15.bmp','blanco16.bmp','blanco17.bmp','blanco18.bmp',...

'blanco19.bmp','blanco20.bmp','blanco21.bmp','blanco22.bmp','blanco23.bmp','blanco24.bmp'};

% Configuración de parámetros
k_min=1;
k_max=6;
color_id='W';
replication=3;
%[color_mixture] = Copy_of_Color_Mixture_Pallete(H,k_min,
k_max,regularize_cov,stop_th,covoption,replication,color_id);
[color_mixture] = Color_Mixture_Pallete(H,k_min,
k_max,regularize_cov,stop_th,covoption,replication,color_id);
close all;

```

A.10 Código 10: Fichero 1

```

%% LLamadas a las funciones de entrenamiento del color
%Parámetros de la función
covoption=0;
regularize_cov=0;
stop_th=0.02;
%LLamada a los ficheros
disp('LLamada a AZUL')
AZUL(covoption, regularize_cov, stop_th);
disp('LLamada a BLANCO')
BLANCO(covoption, regularize_cov, stop_th);
disp('LLamada a DBROWN')
DBROWN(covoption, regularize_cov, stop_th);
disp('LLamada a LBROWN')
LBROWN(covoption, regularize_cov, stop_th);
disp('LLamada a NEGRO')
NEGRO(covoption, regularize_cov, stop_th);
disp('LLamada a RED')
RED(covoption, regularize_cov, stop_th);

```

A.11 Código 11

```

function [TP,FP,FN,TN] = tabla_resultados(ref_seg,masc)
%Función que se encarga de obtener los parámetros TP,FP,FN,TN
%
% Entradas:
% ref_seg = máscara realizada por el especialista
% masc = máscara realizada por el algoritmo
%
%Salidas
%TP: Verdaderos Positivos
%FP: Falsos Positivos
%TN: Verdaderos Negativos
%FN: Falsos Negativos

```

```

% Verdaderos Positivos
TP =ref_seg&mask;
TP=sum(TP(:));
% Falsos Positivos
FP = ~ref_seg&mask;
FP=sum(FP(:));
% Falsos Negativos
FN = ref_seg&~mask;
FN=sum(FN(:));
% Verdaderos Negativos
TN=~ref_seg&~mask;
TN=sum(TN(:));
end

```

A.12 Código 12: Fichero 2

```

%Fichero de llamada a Detect_Colors, en el que asimismo, se invoca a la
%función tabla_resultados, que se encarga de obtener los valores que
%conforman la Tabla 4-1

%Se lee la imagen deseada
img=imread('gzl42.jpg');
%Se lee la máscara para dicha imagen
seg=imread('gzl42_mask_seg.BMP');
%Se pasa a binaria la imagen
seg=im2bw(seg);
%Se establecen parámetros y se llama a la función que detecta colores
step=1;
[color_mask, detected_colors] = Detect_Colors( img, step, seg);
%% Se extraen los resultados
% Máscaras
% color_mask==1 --> BG
% color_mask==2 --> DB
% color_mask==3 --> LB
% color_mask==4 --> B
% color_mask==5 --> W
% color_mask==6 --> R

mask=color_mask==1;
load('gzl42_negro.mat');
ref_seg=gzl42_negro;
[TP,FP,FN,TN] = tabla_resultados(ref_seg,mask);

```

A.13 Código 13: Fichero 3

```

%Fichero de que extrae de excell el vector de colores detectados por
%el algoritmo para cada color.

negrom1=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'B4:B63');
negrom2=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'B70:B113');

```

```

negrom3=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'B120:B141');
negrom=[negrom1;negrom2;negrom3];

azulm1=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'C4:C63');
azulm2=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'C70:C113');
azulm3=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'C120:C141');
azulm=[azulm1;azulm2;azulm3];

darkm1=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'D4:D63');
darkm2=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'D70:D113');
darkm3=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'D120:D141');
darkm=[darkm1;darkm2;darkm3];

lightm1=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'E4:E63');
lightm2=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'E70:E113');
lightm3=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'E120:E141');
lightm=[lightm1;lightm2;lightm3];

redm1=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'F4:F63');
redm2=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'F70:F113');
redm3=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'F120:F141');
redm=[redm1;redm2;redm3];

whitem1=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'G4:G63');
whitem2=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'G70:G113');
whitem3=xlsread('RESULTADOS DE DETECTED COLORS.xlsx', 1, 'G120:G141');
whitem=[whitem1;whitem2;whitem3];

```

A.14 Código 14: Fichero 4

```

%Fichero de que extrae de excell el vector de colores detectados por
%el dermatólogo para cada color.

negrod1=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'B4:B63');
negrod2=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'B70:B113');
negrod3=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'B120:B141');
negrod=[negrod1;negrod2;negrod3];

azuld1=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'C4:C63');
azuld2=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'C70:C113');
azuld3=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'C120:C141');
azuld=[azuld1;azuld2;azuld3];

darkd1=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'D4:D63');
darkd2=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'D70:D113');
darkd3=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'D120:D141');
darkd=[darkd1;darkd2;darkd3];

lightd1=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'E4:E63');
lightd2=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'E70:E113');
lightd3=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'E120:E141');
lightd=[lightd1;lightd2;lightd3];

redd1=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'F4:F63');
redd2=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'F70:F113');
redd3=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'F120:F141');
redd=[redd1;redd2;redd3];

```

```
whited1=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'G4:G63');  
whited2=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'G70:G113');  
whited3=xlsread('COMPARACIÓN ALGORITMO Y ESPECIALISTA.xlsx', 1, 'G120:G141');  
whited=[whited1;whited2;whited3];
```

ANEXO B: DATOS RELEVANTES

En este Anexo se presentan las diferentes tablas en formato Excel utilizadas para obtener los resultados experimentales mostrados en el capítulo 4. Se muestran para poder entender mejor el procedimiento seguido para obtener los mismos.

B.1 Tabla 1: Colores detectados por el algoritmo

	EVALUACIÓN ALGORITMO					
	Negro	Azul	Dark	Light	Red	White
MELANOMAS						
'Aal002'						
'Adl301'						
'Adl305'						
'Adl389'	0	1	1	0	0	0
'Ael490'	1	1	1	1	0	0
'Ael496'						
'Ael504'						
'FAL066'	1	1	1	0	0	0
'FBL020'						
'FCL058'						
'FCL068'	1	1	0	0	0	0
'FFL020'	1	1	1	0	0	0
'Fcl026'	1	1	1	0	0	0
'Fcl040'	1	1	1	0	0	0
'Ffl080'	1	0	1	0	0	0
'Ffl092'	1	1	1	0	0	0
'Fll027'	1	0	0	0	0	0
'Fll069'						
'Fll071'	1	0	1	0	0	0
'Fll126'	1	1	1	0	0	0
'Gal039'	1	1	1	0	0	0
'Gal211'	1	0	1	0	0	0
'Gal220'	1	1	1	1	0	0
'Gal240'	1	0	1	0	0	0
'Gal257'	1	1	0	0	0	0
'Gal262'	1	0	1	0	0	0
'Gcl039'						
'Gcl052'						
'Gdl014'	1	0	1	0	0	0
'Gdl023'	1	1	0	0	0	0
'Gdl037'	1	1	0	0	0	0
'NGL010'	1	1	1	0	0	0

'NGL025'						
'Nal018'	1	1	0	0	0	0
'Nal020'	0	1	1	0	1	0
'Nal025'						
'Nal047'	1	1	0	0	0	0
'Nal049'	1	1	1	0	0	0
'Nal066'	1	1	1	0	0	0
'Nal075'	0	0	1	1	0	0
'Nal077'						
'Nbl001'	1	1	1	0	0	0
'Nbl006'	1	1	1	1	0	0
'Nbl015'	1	0	1	0	0	0
'Nbl017'	1	1	1	0	0	0
'Nbl022'	1	0	0	0	0	0
'Nbl030'	1	1	0	0	0	0
'Nbl034'	1	1	1	0	0	0
'Nbl042'	1	1	0	0	1	0
'Nbl048'	1	1	0	0	0	0
'Nbl060'	1	1	1	0	0	0
'Nbl063'	1	1	0	0	0	0
'Ngl002'	1	0	1	0	0	0
'Ngl004'	1	1	1	0	0	0
'Ngl015'	1	1	1	0	0	0
'Ngl017'						
'Ngl019'						
'Ngl021'	1	1	1	0	0	0
'Ngl027'	1	0	1	1	0	0
'Ngl039'						
'Ngl041'	1	1	1	0	1	0
'Ngl055'						
'Ngl066'	1	1	1	0	0	0
'Ngl068'	0	0	1	0	0	0
'Ngl100'	1	1	1	0	0	0
'Nil031'	1	1	1	0	0	0
'Nil064'	1	0	1	0	0	0
'Nil066'	1	1	0	0	0	0
'Nil078'						
'Nil084'	1	1	1	0	0	0
'Nil103'	1	1	0	0	0	0
'NII002'						
'Nml020'	1	1	1	0	0	0
'Nml073'	1	1	0	0	0	0
'Nml077'	0	1	1	1	0	0
'Nml081'	1	1	1	1	0	0
'gzl04'						
'nal095'	0	0	1	1	1	0

'nfl070'	0	1	1	1	0	0
'nfl072'	1	1	1	1	0	0

CLARK	Negro	Azul	Dark	Light	Red	White
'Aal010'	1	1	1	0	0	0
'Aal012'	1	1	1	0	0	0
'Abl135'	1	1	1	1	0	0
'Abl137'	1	0	1	0	0	0
'Abl139'	1	1	1	0	0	0
'Abl141'	1	0	1	1	0	0
'Abl151'	0	0	1	1	0	0
'Abl157'	1	0	1	0	0	0
'Abl161'	1	1	1	1	0	0
'Abl167'	1	0	1	1	0	0
'Abl168'	1	1	1	0	0	0
'Abl171'	1	1	1	0	0	0
'Abl175'	1	1	1	0	0	0
'Abl179'	1	1	1	1	0	0
'Abl185'						
'Abl195'						
'Acl215'	1	0	1	0	0	0
'Acl219'	0	1	1	1	0	0
'Acl239'	1	1	1	0	0	0
'Acl281'						
'Acl283'	0	0	1	1	0	0
'Acl287'						
'Adl311'	1	0	1	1	1	0
'Adl340'	1	0	1	1	0	0
'Adl391'	1	1	1	1	0	0
'Ael419'	1	1	1	0	0	0
'Ael452'	1	1	1	1	0	0
'Ael471'	1	1	1	1	0	0
'FBL062'	1	1	0	0	0	0
'FCL018'						
'Fbl006'						
'Fbl068'						
'Fbl074'	1	1	1	0	0	0
'Ffl078'	1	1	1	1	0	0
'Fgl068'	1	1	1	0	0	0
'Fhl066'	1	0	1	0	0	0
'Fll023'	1	1	1	0	0	0
'Fll091'	1	1	1	0	0	0
'Gal230'	1	1	1	1	0	0

'Nbl076'	1	0	1	1	0	0
'Nbl086'	1	0	1	1	0	0
'Nbl098'	1	0	1	0	0	0
'Ncl006'	1	0	1	0	0	0
'Ncl030'	1	0	1	1	0	0
'Ncl044'	1	0	1	1	0	0
'Ncl048'	1	0	1	0	0	0
'Ncl063'	1	0	1	1	0	0
'Ncl076'						
'Ndl033'						
'Ndl039'						
'Ndl040'	1	1	1	0	0	0
'Nll024'	1	0	1	1	0	0
'Nll060'	1	0	1	0	0	0
'Nll072'	1	0	1	1	0	0
'Nll098'						

BLUE NEVUS	Negro	Azul	Dark	Light	Red	White
'Ael442'	1	1	0	0	0	0
'Ael444'	1	1	1	0	0	0
'Ael446'	0	1	1	0	1	0
'Ael508'	1	1	0	0	0	0
'Ael510'	1	1	0	0	0	0
'Ael512'	1	1	0	0	0	0
'FBL032'	1	0	0	0	0	0
'FDL026'	1	1	0	0	0	0
'Fcl088'	1	1	0	0	0	0
'Fgl063'	1	1	0	0	0	0
'Fgl090'	1	1	0	0	0	0
'Ggl004'	0	1	0	0	0	0
'Ggl006'	1	1	0	0	0	0
'Nfl046'	0	1	0	0	0	0
'Nfl048'	1	1	0	0	0	0
'Nfl050'	1	1	0	0	0	0
'Ngl090'	0	1	0	0	0	0
'Nml087'	0	1	0	0	0	0
'Nml089'	1	1	0	0	0	0
'gzl24'	0	1	0	0	0	0
'gzl42'	1	1	0	0	0	0
'gzl88'	0	1	0	0	0	0

B.2 Tabla 2: Valores totales de los parámetros TP, TN, FP y FN , y para cada uno de los colores. Valores para obtener Tabla 4-1.

B.2.1 Color Azul-Gris

	BG	TP	TN	FP	FN
MEL	'Aal002'				
	'Adl301'				
	'Adl305'				
	'Adl389'	27989	311398	53761	68
	'Ael490'	852	388224	3208	932
	'Ael496'				
	'Ael504'				
	'FAL066'	8952	369764	9136	5364
	'FBL020'				
	'FCL058'				
	'FCL068'	742	373087	3639	15748
	'FFL020'2				
	'Fcl026'	1246	383547	5146	3277
	'Fcl040'	590	380590	11154	882
	'Ffl080'				
	'Ffl092'	13028	363746	10061	6381
	'Fll027'				
	'Fll069'				
	'Fll071'				
	'Fll126'	1693	386577	4539	407
	'Gal039'				
	'Gal211'				
	'Gal220'	444	382353	3876	6543
	'Gal240'				
	'Gal257'	342	372043	4263	16568
	'Gal262'				
	'Gcl039'				
	'Gcl052'				
	'Gdl014'				
	'Gdl023'	15535	340137	16211	21333
	'Gdl037'				
	'NGL010'	176	386562	5480	998
	'NGL025'				
	'Nal018'	20624	353148	13804	5640
	'Nal020'	2449	348438	42328	1
	'Nal025'				
	'Nal047'	4448	348338	6415	34015
	'Nal049'	8162	364174	16471	4409
	'Nal066'	10367	334313	16695	31841
	'Nal075'				
	'Nal077'				

'Nbl001'	801	344946	45414	2055
'Nbl006'	399	378641	4039	10137
'Nbl015'				
'Nbl017'	1064	375983	4541	11628
'Nbl022'				
'Nbl030'	32209	318671	16997	25339
'Nbl034'	47194	297747	27413	20862
'Nbl042'	9019	354708	6091	23398
'Nbl048'	39604	316529	10570	26513
'Nbl060'	39513	286755	31621	35327
'Nbl063'	32072	338869	4101	18174
'Ngl002'				
'Ngl004'	2549	371381	6414	12872
'Ngl015'	18080	361167	2139	11830
'Ngl017'				
'Ngl019'				
'Ngl021'	3212	319749	4772	65483
'Ngl027'				
'Ngl039'				
'Ngl041'	1802	385264	4524	1626
'Ngl055'				
'Ngl066'	7814	355707	9549	20146
'Ngl068'				
'Ngl100'	109	382900	9324	883
'Nil031'	601	386975	3965	1675
'Nil064'				
'Nil066'	5383	299376	11158	77299
'Nil078'				
'Nil084'	711	369892	8480	14133
'Nil103'	27938	302510	37926	24842
'Nll002'				
'Nml020'	10926	331429	32032	18829
'Nml073'	39697	274156	13137	66226
'Nml077'	2928	384702	712	4874
'Nml081'	32993	343621	6631	9971
'gzl04'				
'nal095'				
'nfl070'				
'nfl072'	329	378913	13771	203
'Aal010'	210	364913	3827	24266
'Aal012'	1572	374084	3012	14548
'Abl135'				
'Abl137'				
'Abl139'	1638	376349	14331	898
'Abl141'				
'Abl151'				

CLARK

'Abl157'				
'Abl161'				
'Abl167'				
'Abl168'	15	387593	4075	1533
'Abl171'				
'Abl175'				
'Abl179'				
'Abl185'				
'Abl195'				
'Acl215'				
'Acl219'				
'Acl239'				
'Acl281'				
'Acl283'				
'Acl287'				
'Adl311'				
'Adl340'				
'Adl391'				
'Ael419'				
'Ael452'				
'Ael471'				
'FBL062'				
'FCL018'				
'Fbl006'				
'Fbl068'				
'Fbl074'	695	380596	4783	7142
'Ffl078'				
'Fgl068'				
'Fhl066'				
'Fll023'				
'Fll091'				
'Gal230'				
'Nbl076'				
'Nbl086'				
'Nbl098'				
'Ncl006'				
'Ncl030'				
'Ncl044'				
'Ncl048'				
'Ncl063'				
'Ncl076'				
'Ndl033'				
'Ndl039'				
'Ndl040'	104	376091	10277	6744
'Nll024'				
'Nll060'				
'Nll072'				

	'NII098'				
BLUE	'Ael442'	31	363961	4383	24841
	'Ael444'	3695	385929	3229	363
	'Ael446'	5061	381487	6381	287
	'Ael508'	4451	385092	2189	1484
	'Ael510'	53847	292331	45151	1887
	'Ael512'	9846	374565	7475	1330
	'FBL032'				
	'FDL026'	51162	291091	36739	14224
	'Fcl088'	5182	380388	6834	812
	'Fgl063'	18363	311782	15361	47710
	'Fgl090'	11171	350294	10338	21413
	'Ggl004'	26066	353202	13158	22
	'Ggl006'	9755	363846	8078	10769
	'Nfl046'	53526	317126	22190	374
	'Nfl048'	64080	305277	20555	3304
	'Nfl050'	33912	276599	65157	17548
	'Ngl090'	56975	300669	35480	92
	'Nml087'	5944	378864	8408	0
	'Nml089'	30571	333865	14687	14093
	'gzl24'	20994	360330	11124	0
	'gzl42'	784	378627	6236	7569
	'gzl88'	43720	333440	15162	126
TOTAL		987956	23925421	940128	882111

B.2.2 Color Marrón Oscuro

	DB	TP	TN	FP	FN
MEL	'Aal002'				
	'Adl301'				
	'Adl305'				
	'Adl389'				
	'Ael490'	5544	370816	6582	10274
	'Ael496'				
	'Ael504'				
	'FAL066'	4535	347200	10720	30761
	'FBL020'				
	'FCL058'				
	'FCL068'				
	'FFL020'2	3624	328086	9202	52304
	'Fcl026'	7763	310928	14161	60364
	'Fcl040'	7308	339745	16745	29418
	'Ffl080'	7851	359229	11399	14737
	'Ffl092'	912	357712	8169	26423
	'FII027'				
	'FII069'				

'FII071'	14635	242389	34286	101906
FII126'	3966	363930	10822	14498
'Gal039'	1032	340099	28012	24073
'Gal211'	11427	327597	28611	25581
'Gal220'	7542	352445	14310	18919
'Gal240'	4747	356063	11641	20765
'Gal257'				
'Gal262'	3638	317061	48404	24113
'Gcl039'				
'Gcl052'				
'Gdl014'	26	362959	7212	23019
'Gdl023'				
Gdl037'				
'NGL010'	7154	336144	25871	24047
'NGL025'				
'Nal018'				
'Nal020'	5706	358793	26831	1886
'Nal025'				
'Nal047'				
'Nal049'	24293	298781	37476	32666
'Nal066'	1989	344163	10996	36068
'Nal075'	2694	370910	17842	1770
'Nal077'				
'Nbl001'	1826	330259	47717	13414
'Nbl006'	331	365337	8395	19153
'Nbl015'	12597	281632	42484	56503
'Nbl017'	2995	308643	9485	72093
'Nbl022'				
'Nbl030'				
'Nbl034'	540	354256	10692	27728
'Nbl042'				
'Nbl048'				
'Nbl060'	3776	335758	14942	38740
'Nbl063'				
'Ngl002'	1650	353034	12045	26487
'Ngl004'	23206	284387	27831	57792
'Ngl015'	9172	330069	23840	30135
'Ngl017'				
'Ngl019'				
'Ngl021'	34	357568	7706	27908
'Ngl027'	19946	308550	22402	42318
'Ngl039'				
'Ngl041'	8295	355691	25953	3277
'Ngl055'				
'Ngl066'	166	369156	11504	12390
'Ngl068'	14120	245378	132418	1300
'Ngl100'	24807	247654	40742	80013

	'Nil031'	1723	375381	12827	3285
	'Nil064'	10552	343799	36629	2236
	'Nil066'				
	'Nil078'				
	'Nil084'	4553	322037	12929	53697
	'Nil103'				
	'Nil002'				
	'Nml020'	1924	336211	32313	22768
	'Nml073'				
	'Nml077'	1670	363633	27772	141
	'Nml081'	6649	347885	34371	4311
	'gzl04'				
	'nal095'	6029	351489	35067	631
	'nfl070'	191	372927	15624	4474
	'nfl072'	19323	321076	22216	30601
CLARK	'Aal010'	393	377246	4318	11259
	'Aal012'	4090	362444	9981	16701
	'Abl135'	11095	341109	11783	29229
	'Abl137'	17545	267593	18459	89619
	'Abl139'	32050	292126	16762	52278
	'Abl141'	151	386481	2179	4405
	'Abl151'	3550	366137	18835	4694
	'Abl157'	71749	250616	26320	44531
	'Abl161'	8164	361726	11098	12228
	'Abl167'	2651	366935	13761	9869
	'Abl168'	11473	361880	12088	7775
	'Abl171'	10410	306699	71333	4774
	'Abl175'	20480	343608	20224	8904
	'Abl179'	25578	328733	14051	24854
	'Abl185'				
	'Abl195'				
	'Acl215'	9319	354832	15064	14001
	'Acl219'	3656	374087	14781	692
	'Acl239'	36245	254218	18618	84135
	'Acl281'				
	'Acl283'	14027	296294	77358	5537
	'Acl287'				
	'Adl311'	12644	360294	16538	3740
	'Adl340'	12959	319841	43563	16853
	'Adl391'	53	369938	3094	20131
	'Ael419'	247	382187	4814	5968
	'Ael452'	3839	355547	4809	29021
	'Ael471'	9214	349733	25015	9254
	'FBL062'				
	'FCL018'				
	'Fbl006'				

	'Fbl068'				
	'Fbl074'	1442	354185	16284	21305
	'Ffl078'	6881	344783	10501	31051
	'Fgl068'				
	'Fhl066'	2571	363708	23246	3691
	'Fll023'	2348	319540	29287	42041
	'Fll091'	6266	314155	12940	59855
	'Gal230'	616	375497	3030	14073
	'Nbl076'	38651	330007	15937	8621
	'Nbl086'	10345	359146	9343	14382
	'Nbl098'	3920	358523	9165	21608
	'Ncl006'	5195	364055	9153	14813
	'Ncl030'	14551	315606	54524	8535
	'Ncl044'	1750	370939	4645	15882
	'Ncl048'	1296	376800	5248	9872
	'Ncl063'	2179	382266	5326	3445
	'Ncl076'				
	'Ndl033'				
	'Ndl039'				
	'Ndl040'	2988	348349	7873	34006
	'Nll024'	7013	364731	12516	8956
	'Nll060'	8608	305039	34560	45009
	'Nll072'	66838	254610	55642	16126
	'Nll098'				
BLUE	'Ael442'				
	'Ael444'				
	'Ael446'				
	'Ael508'				
	'Ael510'				
	'Ael512'				
	'FBL032'				
	'FDL026'				
	'Fcl088'				
	'Fgl063'				
	'Fgl090'				
	'Ggl004'				
	'Ggl006'				
	'Nfl046'				
	'Nfl048'				
	'Nfl050'				
	'Ngl090'				
	'Nml087'				
	'Nml089'				
	'gzl24'				
	'gzl42'				
	'gzl88'				
TOTAL		807501	29079103	1841262	2088710

B.2.3 Color Marrón Claro

	LB	TP	TN	FP	FN
MEL	'Aal002'				
	Adl301'				
	'Adl305'				
	'Adl389'				
	'Ael490'	2532	366934	21243	2507
	'Ael496'				
	'Ael504'				
	'FAL066'				
	'FBL020'				
	'FCL058'				
	'FCL068'				
	FFL020'2				
	'Fcl026'				
	'Fcl040'				
	'Ffl080'				
	'Ffl092'				
	'Fll027'				
	'Fll069'				
	'Fll071'				
	Fll126'				
	'Gal039'				
	'Gal211'				
	'Gal220'	7152	362832	13774	9458
	'Gal240'				
	'Gal257'				
	'Gal262'				
	'Gcl039'				
	'Gcl052'				
	'Gdl014'				
	'Gdl023'				
	Gdl037'				
	'NGL010'				
	'NGL025'				
	'Nal018'				
	'Nal020'				
	'Nal025'				
	'Nal047'				
	'Nal049'				
	'Nal066'				
	'Nal075'	85769	257620	32416	17411
	'Nal077'				
	'Nbl001'				
	'Nbl006'	8668	373074	7254	4220
	'Nbl015'				

	'Nbl017'				
	'Nbl022'				
	'Nbl030'				
	'Nbl034'				
	'Nbl042'				
	'Nbl048'				
	'Nbl060'				
	'Nbl063'				
	'Ngl002'				
	'Ngl004'				
	'Ngl015'				
	'Ngl017'				
	'Ngl019'				
	'Ngl021'				
	'Ngl027'	3291	361489	17163	11273
	'Ngl039'				
	'Ngl041'				
	'Ngl055'				
	'Ngl066'				
	'Ngl068'				
	'Ngl100'				
	'Nil031'				
	'Nil064'				
	'Nil066'				
	'Nil078'				
	'Nil084'				
	'Nil103'				
	'Nll002'				
	'Nml020'				
	'Nml073'				
	'Nml077'	11681	345352	14179	22004
	'Nml081'	9457	330668	16676	36415
	'gzl04'				
	'nal095'	6092	339463	7901	39760
	'nfl070'	50041	288020	41548	13607
	'nfl072'	5518	358441	17539	11718
CLARK	'Aal010'				
	'Aal012'				
	'Abl135'	1350	368184	18812	4870
	'Abl137'				
	'Abl139'				
	'Abl141'				
	'Abl151'	27298	330155	18809	16954
	'Abl157'				
	'Abl161'	8100	364634	14970	5512
	'Abl167'	8143	356105	15575	13393
	'Abl168'				

'Abl171'				
'Abl175'				
'Abl179'	6258	361490	13566	11902
'Abl185'				
'Abl195'				
'Acl215'				
'Acl219'	28024	329794	18802	16596
'Acl239'				
'Acl281'				
'Acl283'	22441	273032	25804	71939
'Acl287'				
'Adl311'	11953	349408	13272	18583
'Adl340'	16439	319119	19545	38113
'Adl391'	598	383336	8896	386
'Ael419'				
'Ael452'	4445	367538	17546	3687
'Ael471'	4469	352731	11301	24715
'FBL062'				
'FCL018'				
'Fbl006'				
'Fbl068'				
'Fbl074'				
'Ffl078'	7882	364614	12750	7970
'Fgl068'				
'Fhl066'				
'Fll023'				
'Fll091'				
'Gal230'	942	384731	6785	758
'Nbl076'	2158	368912	12388	9758
'Nbl086'	31922	330882	18678	11734
'Nbl098'				
'Ncl006'				
'Ncl030'	10925	310016	18180	54095
'Ncl044'				
'Ncl048'				
'Ncl063'	96389	266228	23592	7007
'Ncl076'				
'Ndl033'				
'Ndl039'				
'Ndl040'				
'Nll024'	2031	375264	8045	7876
'Nll060'				
'Nll072'	27369	307870	17914	40063
'Nll098'				
'Ael442'				
'Ael444'				

BLUE

'Ael446'				
'Ael508'				
'Ael510'				
'Ael512'				
'FBL032'				
'FDL026'				
'Fcl088'				
'Fgl063'				
'Fgl090'				
'Ggl004'				
'Ggl006'				
'Nfl046'				
'Nfl048'				
'Nfl050'				
'Ngl090'				
'Nml087'				
'Nml089'				
'gzl24'				
'gzl42'				
'gzl88'				
TOTAL	509337	10247936	504923	534284

B.2.4 Color Negro

	NEGRO	TP	TN	FP	FN
MEL	'Aal002'				
	'Adl301'				
	'Adl305'				
	'Adl389'				
	'Ael490'	296	387690	5083	147
	'Ael496'				
	'Ael504'				
	'FAL066'	18421	335529	37859	1407
	'FBL020'				
	'FCL058'				
	'FCL068'	18514	332294	40937	1471
	'FFL020'2	4327	317922	70949	18
	'Fcl026'	5231	304989	82857	139
	'Fcl040'	1465	358298	33453	0
	'Ffl080'	5317	370709	17190	0
	'Ffl092'	1681	345363	46074	98
	'Fll027'	34836	252820	105511	49
	'Fll069'				
	'Fll071'	1646	273766	117804	0
	'Fll126'	62405	304225	25599	987
	'Gal039'	13030	352233	26989	964

'Gal211'	3506	337425	52033	252
'Gal220'	11194	355717	24884	1421
'Gal240'	19054	346232	27040	890
'Gal257'	30658	309572	52359	627
'Gal262'	17441	314607	60538	630
'Gcl039'				
'Gcl052'				
'Gdl014'	1155	332230	59828	3
'Gdl023'	16738	350705	25656	117
Gdl037'				
'NGL010'	484	359480	33252	0
'NGL025'				
'Nal018'	2173	384410	6472	161
'Nal020'				
'Nal025'				
'Nal047'	14940	312217	66059	0
'Nal049'	790	347623	44802	1
'Nal066'	284	315013	77919	0
'Nal075'				
'Nal077'				
'Nbl001'				
'Nbl006'	41	363786	29389	0
'Nbl015'				
'Nbl017'	4187	290313	98715	1
'Nbl022'	46663	220903	125640	10
'Nbl030'	2826	355885	34423	82
'Nbl034'	1235	341965	50016	0
'Nbl042'	348	363372	29496	0
'Nbl048'	22510	286847	83791	68
'Nbl060'				
'Nbl063'	9489	317285	66442	0
'Ngl002'	8758	320523	63927	8
'Ngl004'	30	308773	84413	0
'Ngl015'	1519	337163	54534	0
'Ngl017'				
'Ngl019'				
'Ngl021'	4585	273545	115079	7
'Ngl027'	2742	347124	43350	0
'Ngl039'				
'Ngl041'				
'Ngl055'				
'Ngl066'	1007	344910	47299	0
'Ngl068'				
'Ngl100'				
'Nil031'	1263	382715	9207	31
'Nil064'				

	'Nil066'	5276	268653	119285	2
	'Nil078'				
	'Nil084'	429	305518	87269	0
	'Nil103'				
	'NII002'				
	'Nml020'	709	336788	55719	0
	'Nml073'	1452	290969	100767	28
	'Nml077'				
	'Nml081'				
	'gzi04'				
	'nal095'				
	'nfl070'				
	'nfl072'				
CLARK	'Aal010'	257	350146	42813	0
	'Aal012'				
	'Abl135'				
	'Abl137'	4278	301699	87213	26
	'Abl139'				
	'Abl141'	7008	380687	5421	100
	'Abl151'				
	'Abl157'	1436	345566	46170	44
	'Abl161'				
	'Abl167'	28	386764	6424	0
	'Abl168'				
	'Abl171'				
	'Abl175'				
	'Abl179'				
	'Abl185'				
	'Abl195'				
	'Acl215'	581	377748	14850	37
	'Acl219'				
	'Acl239'	2075	313602	77539	0
	'Acl281'				
	'Acl283'				
	'Acl287'				
	'Adl311'				
	'Adl340'				
	'Adl391'	140	377592	15480	4
	'Ael419'	5123	374665	13400	28
	'Ael452'	2730	366082	24335	69
	'Ael471'				
	'FBL062'	4204	364592	24420	0
	'FCL018'				
	'Fbl006'				
	'Fbl068'				
	'Fbl074'				
	'Ffl078'	432	368795	23989	0

	'Fgl068'	9568	377927	5583	138
	'Fhl066'				
	'Fll023'	471	346078	46666	1
	'Fll091'	712	329283	63221	0
	'Gal230'	4909	373973	14328	6
	'Nbl076'				
	'Nbl086'				
	'Nbl098'	302	371149	21764	1
	'Ncl006'				
	'Ncl030'	183	387555	5455	23
	'Ncl044'	1991	374781	16444	0
	'Ncl048'	227	381603	11386	0
	'Ncl063'				
	'Ncl076'				
	'Ndl033'				
	'Ndl039'				
	'Ndl040'	643	356078	36495	0
	'Nll024'	439	383032	9535	210
	'Nll060'				
	'Nll072'				
	'Nll098'				
BLUE	'Ael442'	996	357392	34816	12
	'Ael444'				
	'Ael446'				
	'Ael508'				
	'Ael510'				
	'Ael512'				
	'FBL032'				
	'FDL026'				
	'Fcl088'				
	'Fgl063'	1025	307196	84936	59
	'Fgl090'	190	368514	24378	134
	'Ggl004'				
	'Ggl006'				
	'Nfl046'				
	'Nfl048'				
	'Nfl050'				
	'Ngl090'				
	'Nml087'				
	'Nml089'				
	'gzl24'				
	'gzl42'	344	383819	9049	4
	'gzl88'				
TOTAL		450947	23464424	3206018	10515

B.2.5 Color Rojo

	RED	TP	TN	FP	FN
MEL	'Aal002'				
	Adl301'				
	'Adl305'				
	'Adl389'				
	'Ael490'				
	'Ael496'				
	'Ael504'				
	'FAL066'				
	'FBL020'				
	'FCL058'				
	'FCL068'				
	FFL020'2				
	'Fcl026'				
	'Fcl040'				
	'Ffl080'				
	'Ffl092'				
	'Fll027'				
	'Fll069'				
	'Fll071'				
	Fll126'				
	'Gal039'				
	'Gal211'				
	'Gal220'				
	'Gal240'				
	'Gal257'				
	'Gal262'				
	'Gcl039'				
	'Gcl052'				
	'Gdl014'				
	'Gdl023'				
	Gdl037'				
	'NGL010'				
	'NGL025'				
	'Nal018'				
	'Nal020'				
	'Nal025'				
	'Nal047'				
	'Nal049'				
	'Nal066'				
	'Nal075'				
	'Nal077'				
	'Nbl001'				
	'Nbl006'				
	'Nbl015'				

'Nbl017'				
'Nbl022'				
'Nbl030'				
'Nbl034'				
'Nbl042'	528	389125	2490	1073
'Nbl048'				
'Nbl060'				
'Nbl063'				
'Ngl002'				
'Ngl004'				
'Ngl015'				
'Ngl017'				
'Ngl019'				
'Ngl021'				
'Ngl027'				
'Ngl039'				
'Ngl041'	2339	387627	1269	1981
'Ngl055'				
'Ngl066'				
'Ngl068'				
'Ngl100'				
'Nil031'				
'Nil064'				
'Nil066'				
'Nil078'				
'Nil084'				
'Nil103'				
'Nil002'				
'Nml020'				
'Nml073'				
'Nml077'				
'Nml081'				
'gzl04'				
'nal095'	13171	359199	20025	821
'nfl070'				
'nfl072'				
'Aal010'				
'Aal012'				
'Abl135'				
'Abl137'				
'Abl139'				
'Abl141'				
'Abl151'				
'Abl157'				
'Abl161'				
'Abl167'				

CLARK

'Abl168'
'Abl171'
'Abl175'
'Abl179'
'Abl185'
'Abl195'
'Acl215'
'Acl219'
'Acl239'
'Acl281'
'Acl283'
'Acl287'
'Adl311'
'Adl340'
'Adl391'
'Ael419'
'Ael452'
'Ael471'
'FBL062'
'FCL018'
'Fbl006'
'Fbl068'
'Fbl074'
'Ffl078'
'Fgl068'
'Fhl066'
'Fll023'
'Fll091'
'Gal230'
'Nbl076'
'Nbl086'
'Nbl098'
'Ncl006'
'Ncl030'
'Ncl044'
'Ncl048'
'Ncl063'
'Ncl076'
'Ndl033'
'Ndl039'
'Ndl040'
'Nll024'
'Nll060'
'Nll072'
'Nll098'
'Ael442'
'Ael444'

BLUE

'Ael446'				
'Ael508'				
'Ael510'				
'Ael512'				
'FBL032'				
'FDL026'				
'Fcl088'				
'Fgl063'				
'Fgl090'				
'Ggl004'				
'Ggl006'				
'Nfl046'				
'Nfl048'				
'Nfl050'				
'Ngl090'				
'Nml087'				
'Nml089'				
'gzl24'				
'gzl42'				
'gzl88'				
TOTAL	16038	1135951	23784	3875

B.2.6 Resultados totales

TP	2771779
TN	87852835
FP	6516115
FN	3519495

B.3 Tabla 3: Valores totales del parámetro TP, de cada color, para el caso de clasificación errónea dado en la Tabla 4-2.

			EVALUACIÓN ALGORITMO			
MELANOMAS	Negro	Azul	Dark	Light	Red	White
'Aal002'						
'Adl301'						
'Adl305'						
'Adl389'			28057		32275	
'Ael490'		443	1784	15818	5039	
'Ael496'						
'Ael504'						

'FAL066'	19828	14316	35296	12516	
'FBL020'					
'FCL058'					
'FCL068'	19985	16490	22263		
'FFL020'	4345		55928	16847	
'Fcl026'	5370	4523	68127	596	
'Fcl040'	1465	1472	36726	12239	4716
'Ffl080'	5317		22588	1792	
'Ffl092'	1779	19409	27335		
'Fll027'	34885		55017	8030	
'Fll069'					
'Fll071'	1646	2116	116541	9892	
'Fll126'	63392	2100	18464	7100	
'Gal039'	13994		25105	34285	
'Gal211'	3758	9725	37008	7269	27142
'Gal220'	12615	6987	26461	16610	
'Gal240'	19944		25512	1668	
'Gal257'	31285	16910	12648	8517	
'Gal262'	18071	7160	27751	40920	
'Gcl039'					
'Gcl052'					
'Gdl014'	1158	1040	23045		
'Gdl023'	16855	36868	944		
'Gdl037'			11077	40702	7799
'NGL010'	484	1174	31201	9361	
'NGL025'					
'Nal018'	2334	26264			
'Nal020'		2450	7592	41928	26596
'Nal025'					
'Nal047'	14940	38463	7320	4420	
'Nal049'	791	12571	56959	13501	1776
'Nal066'	284	42208	38057	6798	
'Nal075'			4464	103180	
'Nal077'					
'Nbl001'		2856	15240	69096	908
'Nbl006'	41	10536	19484	12888	
'Nbl015'		42340	69100	24036	
'Nbl017'	4188	12692	75088	4172	
'Nbl022'	46673	68380	3900		
'Nbl030'	2908	57548	1396		
'Nbl034'	1235	68056	28268	11792	
'Nbl042'	348	32417	4062		1601
'Nbl048'	22578	66117	10940		
'Nbl060'		74840	42516	24548	
'Nbl063'	9489	50246	18498		
'Ngl002'	8766	18776	28137		
'Ngl004'	30	15421	80998	5565	

'Ngl015'	1519	29910	39307	5127	84	
'Ngl017'						
'Ngl019'						
'Ngl021'	4592	68695	27942			
'Ngl027'	2742	5856	62264	14564		
'Ngl039'						
'Ngl041'		3428	11572	16844	4320	
'Ngl055'						
'Ngl066'	1007	27960	12556	14366		
'Ngl068'			15420	134632		
'Ngl100'		992	104820	33480		
'Nil031'	1294	2276	5008	4302		
'Nil064'		7534	12788	43090		
'Nil066'	5278	82682	11243			
'Nil078'						
'Nil084'	429	14844	58250	4640		
'Nil103'		52780		21660	7748	
'Nll002'						
'Nml020'	709	29755	24692	31849	11167	
'Nml073'	1480	105923	5208			
'Nml077'		7802	1811	33685		
'Nml081'		42964	10960	45872		
'gzl04'						
'nal095'			6660	45852	13992	773
'nfl070'			4665	63648		29763
'nfl072'		532	49924	17236		
TOTAL	410274	1296245	1671964	1118429	66105	72280

CLARK	Negro	Azul	Dark	Light	Red	White
'Aal010'	257	24476	11652			
'Aal012'		16120	20791	5264		
'Abl135'			40324	6220		
'Abl137'	4304		107164	1192		
'Abl139'		2536	84328	27376		
'Abl141'	7108		4556			
'Abl151'			8244	44252		
'Abl157'	1480	2632	116280	11916		
'Abl161'			20392	13612		1172
'Abl167'	28		12520	21536		
'Abl168'		1548	19248	6364		
'Abl171'			15184	71124		
'Abl175'			29384	23972		
'Abl179'			50432	18160		

'Abl185'						
'Abl195'						
'Acl215'	618		23320	8144		
'Acl219'			4348	44620		
'Acl239'	2075		120380	6604		
'Acl281'						
'Acl283'			19564	94380		
'Acl287'						
'Adl311'			16384	30536		
'Adl340'			29812	54552		
'Adl391'	144		20184	984		
'Ael419'	5151		6215			
'Ael452'	2799		32860	8132		
'Ael471'			18468	29184		
'FBL062'	4204		19820	368		
'FCL018'						
'Fbl006'						
'Fbl068'						
'Fbl074'		7837	22747	455		
'Ffl078'	432		37932	15852		
'Fgl068'	9706			3822		
'Fhl066'			6262	18100		
'Fll023'	472		44389	25345		
'Fll091'	712		66121	3717		
'Gal230'	4915		14689	1700		
'Nbl076'			47272	11916		
'Nbl086'		313	24727	43656		
'Nbl098'	303		25528	3232		
'Ncl006'			20008	1240		
'Ncl030'	206		23086	65020		
'Ncl044'	1991		17632			
'Ncl048'	227		11168			
'Ncl063'			5624	103396		
'Ncl076'						
'Ndl033'						
'Ndl039'						
'Ndl040'	643	6848	36994			
'Nll024'	649		15969	9907		
'Nll060'			53617	6789		
'Nll072'			82964	67432		
'Nll098'						
TOTAL	48424	62310	1408583	910071	0	1172

BLUE NEVUS

Negro

Azul

Dark

Light

Red

White

'Ael442'	1008	24872	3004	2116		
'Ael444'		4058				
'Ael446'		5348		2746		
'Ael508'		5935		8700		
'Ael510'		55734	14187	11856		
'Ael512'		11176				
'FBL032'		18210	43685			
'FDL026'		65386		22997		
'Fcl088'		5994	3706			
'Fgl063'	1084	66073	33771			
'Fgl090'	324	32584	928			
'Ggl004'		26088				
'Ggl006'		20524				
'Nfl046'		53900				
'Nfl048'		67384				524
'Nfl050'		51460		32088		
'Ngl090'		57067		15418		
'Nml087'		5944		1452		
'Nml089'		44664				
'gzl24'		20994				
'gzl42'	348	8353		182		
'gzl88'		43846				
TOTAL	2764	695594	99281	97555	0	524
	NEGRO	AZUL	DARK	LIGHT	RED	WHITE
TOTAL	461462	2054149	3179828	2126055	66105	73976

B.4 Tabla 4: Colores detectados por el dermatólogo

MELANOMAS	EVALUACIÓN DEL ESPECIALISTA					
	Negro	Azul	Dark	Light	Red	White
'Aal002'	0	1	1	0	0	0
'Adl301'	1	1	1	1	1	0
'Adl305'	1	1	1	0	0	0
'Adl389'	0	1	0	1	0	0
'Ael490'	1	1	1	1	0	0
'Ael496'	1	0	1	1	0	0
'Ael504'	0	1	1	1	1	0
'FAL066'	1	1	1	1	0	0
'FBL020'	1	1	1	1	0	0
'FCL058'	1	0	1	1	0	0
'FCL068'	1	1	1	0	0	0
'FFL020'	1	0	1	1	0	0
'Fcl026'	1	1	1	1	0	0

'Fcl040'	1	1	1	1	0	1
'Ffl080'	1	0	1	1	0	0
'Ffl092'	1	1	1	0	0	0
'Fll027'	1	0	1	1	0	0
'Fll069'	1	1	1	1	0	0
'Fll071'	1	1	1	1	0	0
'Fll126'	1	1	1	1	0	0
'Gal039'	1	0	1	1	0	0
'Gal211'	1	1	1	1	1	0
'Gal220'	1	1	1	1	0	0
'Gal240'	1	0	1	1	0	0
'Gal257'	1	1	1	1	0	0
'Gal262'	1	1	1	1	0	0
'Gcl039'	1	1	1	1	0	0
'Gcl052'	1	1	0	1	1	0
'Gdl014'	1	1	1	0	0	0
'Gdl023'	1	1	1	0	0	0
'Gdl037'	0	0	1	1	1	0
'NGL010'	1	1	1	1	0	1
'NGL025'	0	1	1	1	1	0
'Nal018'	1	1	0	0	0	0
'Nal020'	0	1	1	1	0	1
'Nal025'	1	1	1	0	1	1
'Nal047'	1	1	1	1	0	0
'Nal049'	1	1	1	1	0	1
'Nal066'	1	1	1	1	0	0
'Nal075'	0	0	1	1	0	0
'Nal077'	1	1	1	1	1	1
'Nbl001'	0	1	1	1	0	1
'Nbl006'	1	1	1	1	0	0
'Nbl015'	0	1	1	1	0	0
'Nbl017'	1	1	1	1	0	0
'Nbl022'	1	1	1	0	0	0
'Nbl030'	1	1	1	0	0	0
'Nbl034'	1	1	1	1	0	0
'Nbl042'	1	1	1	0	1	0
'Nbl048'	1	1	1	1	0	0
'Nbl060'	0	1	1	1	0	0
'Nbl063'	1	1	1	0	0	0
'Ngl002'	1	1	1	0	0	0
'Ngl004'	1	1	1	1	0	0
'Ngl015'	1	1	1	1	1	0
'Ngl017'	1	1	1	1	0	0
'Ngl019'	1	1	1	1	1	1
'Ngl021'	1	1	1	0	0	0
'Ngl027'	1	1	1	1	0	0
'Ngl039'	1	1	1	0	0	0

'Ngl041'	0	1	1	1	1	0
'Ngl055'	1	1	1	0	0	0
'Ngl066'	1	1	1	1	0	0
'Ngl068'	0	0	1	1	0	0
'Ngl100'	0	1	1	1	0	0
'Nil031'	1	1	1	1	0	0
'Nil064'	0	1	1	1	1	0
'Nil066'	1	1	1	0	0	0
'Nil078'	1	1	1	1	0	0
'Nil084'	1	1	1	1	0	0
'Nil103'	0	1	0	1	0	1
'Nil002'	1	1	1	1	0	0
'Nml020'	1	1	1	1	1	0
'Nml073'	1	1	1	0	0	0
'Nml077'	0	1	1	1	0	0
'Nml081'	0	1	1	1	0	0
'gzl04'	1	1	1	1	0	0
'nal095'	0	0	1	1	1	1
'nfl070'	0	0	1	1	0	1
'nfl072'	0	1	1	1	0	0

CLARK	Negro	Azul	Dark	Light	Red	White
'Aal010'	1	1	1	0	0	0
'Aal012'	0	1	1	1	0	0
'Abl135'	0	0	1	1	0	0
'Abl137'	1	0	1	1	0	0
'Abl139'	0	1	1	1	0	1
'Abl141'	1	0	1	0	0	0
'Abl151'	0	0	1	1	0	0
'Abl157'	1	1	1	1	0	0
'Abl161'	0	0	1	1	0	1
'Abl167'	1	0	1	1	0	0
'Abl168'	0	1	1	1	0	0
'Abl171'	0	0	1	1	0	0
'Abl175'	0	0	1	1	0	0
'Abl179'	0	0	1	1	0	0
'Abl185'	1	1	1	1	0	0
'Abl195'	0	0	1	1	1	0
'Acl215'	1	0	1	1	0	0
'Acl219'	0	0	1	1	0	0
'Acl239'	1	0	1	1	0	0
'Acl281'	0	0	1	1	0	0

'Acl283'	0	0	1	1	0	0
'Acl287'	0	1	1	1	0	0
'Adl311'	0	0	1	1	0	0
'Adl340'	0	0	1	1	0	0
'Adl391'	1	0	1	1	0	0
'Ael419'	1	0	1	0	0	0
'Ael452'	1	0	1	1	0	0
'Ael471'	0	0	1	1	0	0
'FBL062'	1	0	1	1	0	0
'FCL018'	1	0	1	0	0	0
'Fbl006'	1	0	1	1	0	0
'Fbl068'	1	1	1	1	0	0
'Fbl074'	0	1	1	1	0	0
'Ffl078'	1	0	1	1	0	0
'Fgl068'	1	0	0	1	0	0
'Fhl066'	0	0	1	1	0	0
'Fll023'	1	0	1	1	0	0
'Fll091'	1	0	1	1	0	0
'Gal230'	1	0	1	1	0	0
'Nbl076'	0	0	1	1	0	0
'Nbl086'	0	1	1	1	0	0
'Nbl098'	1	0	1	1	0	0
'Ncl006'	0	0	1	1	0	0
'Ncl030'	1	0	1	1	0	0
'Ncl044'	1	0	1	0	0	0
'Ncl048'	1	0	1	0	0	0
'Ncl063'	0	0	1	1	0	0
'Ncl076'	1	0	1	0	0	0
'Ndl033'	1	0	1	0	0	0
'Ndl039'	0	1	1	0	0	0
'Ndl040'	1	1	1	0	0	0
'Nll024'	1	0	1	1	0	0
'Nll060'	0	0	1	1	0	0
'Nll072'	0	0	1	1	0	0
'Nll098'	0	0	1	1	0	1

BLUE NEVUS	Negro	Azul	Dark	Light	Red	White
'Ael442'	0	1	1	1	0	0
'Ael444'	0	1	0	0	0	0
'Ael446'	0	1	0	1	0	0
'Ael508'	0	1	0	1	0	0
'Ael510'	0	1	1	1	0	0
'Ael512'	0	1	0	0	0	0

'FBL032'	0	1	1	0	0	0
'FDL026'	0	1	0	1	0	0
'Fcl088'	0	1	1	0	0	0
'Fgl063'	1	1	1	0	0	0
'Fgl090'	0	1	1	0	0	0
'Ggl004'	0	1	0	0	0	0
'Ggl006'	0	1	0	0	0	0
'Nfl046'	0	1	0	0	0	0
'Nfl048'	0	1	0	0	0	0
'Nfl050'	0	1	0	1	0	0
'Ngl090'	0	1	0	1	0	0
'Nml087'	0	1	0	1	0	0
'Nml089'	0	1	0	0	0	0
'gzl24'	0	1	0	1	0	0
'gzl42'	1	1	0	1	0	0
'gzl88'	0	1	0	0	0	0

B.5 Tabla 5: Valores totales de las variables TP, TN, FP y FN de cada color, para el caso de la detección de colores en la Tabla 4-3.

	TP	FP	FN	TN
Black	67	42	0	17
BG	68	21	12	25
DB	86	4	20	16
LB	30	2	65	29
Red	3	3	5	115
White	0	0	10	116
TOTAL	254	72	112	318

